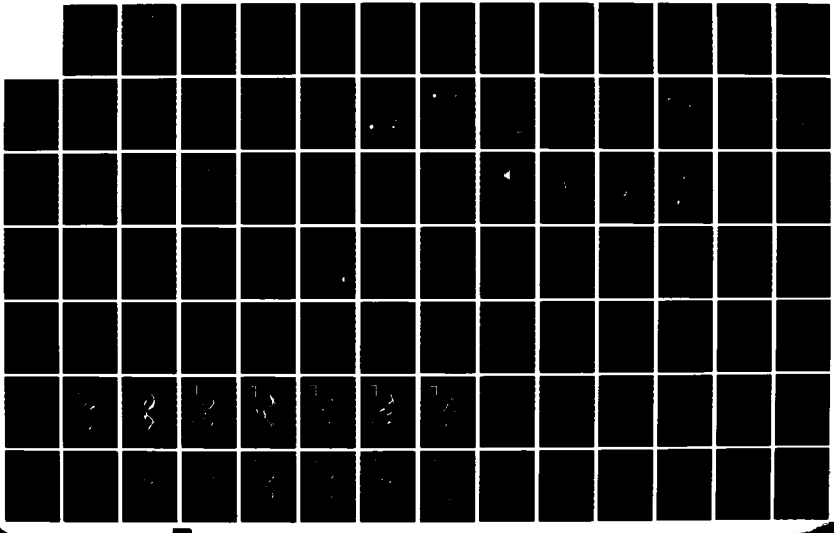
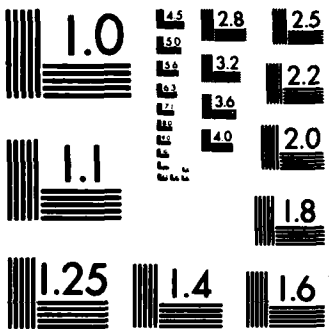
AD-A153 175 MAXIMIZING THE SEMI-MAJOR AXIS FOR A FREELY CONING SOLAR SAILCRAFT(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 1/2

UNCLASSIFIED M R BORJA DEC 84 AFIT/GA/AA/84D-2 F/G 22/3 NL





MAXIMIZING THE SEMI-MAJOR AXIS FOR A FREELY CONING SOLAR SAILCRAFT

THESIS

MARIO REYES BORJA
CAPT USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/AA/84D-2

MAXIMIZING THE SEMI-MAJOR AXIS FOR A FREELY CONING SOLAR SAILCRAFT

THESIS

MARIO REYES BORJA
CAPT USAF

AFIT/GA/AA/84D-2

Approved for public release: distribution unlimited.

AFIT/GA/AA/84D-2

MAXIMIZING THE SEMI-MAJOR AXIS FOR A FREELY CONING SOLAR SAILCRAFT

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements of the Degree of

MASTERS OF SCIENCE IN ASTRONAUTICS

MARIO REYES BORJA, B.S., M.S.

Captain, USAF

December 1984

Approved for public release; distribution unlimited.

Acknowledgements

I dedicate this page to all those wonderful and beautiful people who have been so generous in sharing their precious time with me in my pursuit of this academic endeavor. I extend my sincerest gratitute to Dr. William Weisel, my theses advisor, for being patient, supportive, and understanding throughout my thesis term. To Dr. David Lee and Dr. John J. Jones for unselfishingly providing me the so needed tutorial in numerical methods, I thank you so very much. To Navy Commander Wiegand, I extend my many thanks for the guiding hand in helping me learn the intricacies of the S-Package plotting routines. To Yoshiyaki and Laura Suzuki, special friends of mine who shared the midnight oil with me helping debug my numerous fortran errors, I extend my "Arigato Gozaimasu". To my special cohorts in the ranks, Captains Roy Nici and John Ward, I salute you for all the support you have given me in learning the Vax and providing the needed feedback for understanding astrodynamics. To a very special friend of mine, Henry Baird, who has on many occasions asked the most crucial questions which inevitably guided me back on track of my objective, I take this opportunity to say "thanks a million". And most especilly to my most loving wife and friend, JoJo Ann, who has given me the psychological and administrative support I desperately needed throughout this entire effort, and to my foster children for taking over the chores I have neglected because of this effort, I love you all. To God, who has blessed this whole pursuit for higher education, I thank you with all my heart and soul.

Table of Contents

List of Figures

List of Tables

List of Symbols

Orbital Parameters:

a	Semi-Major Axis
e	Eccentricity
i	Inclination
f	True Anomaly
n	Orbital Mean Motion
TP	Orbital Period
Ω	Longitude of the Ascending Node
μ	Gravitational Parameter (DU^3/TU^2)
Δ	Change in Orbital Parameter

Sailcraft Parameters:

A	Solar Sail Area
A'	Minor Moments of Inertia (principal) about $\bar{b}_1$ and $\bar{b}_2$ Axis.
C'	Moment of Inertia about $\bar{b}_3$ Axis.
κ	Sail Reflectivity Constant; $(0.0 < \kappa < 1.0)$
D	Sailcraft Acceleration Constant Defined in Section 2.4.
T	Thrust Due to Solar Radiation Pressure
ω	Spin Rate of Spacecraft about $\bar{b}_3$ Axis.

Sail Orientation Parameters:

η	Angular Momentum Orientation Angle wrt $\bar{K}$ Axis
ζ	Angular Momentum Orientation Angle wrt $\bar{I}$ Axis
ν	Precession Rate
θ	Nutation Angle or Coning Angle

ϕ	Phase Angle Associated with the True Anomaly
ψ	Phase Angle Associtated with the Sail Precession

Coordinates/Reference Frames:

b_i	Body-Fixed Axes of Solar Sailcraft; i = 1,2,3.
i,j,k	Orbital Plane Inertial Frame.
I,J,K	Body-centric Inertial Frame.
$\bar{U},\bar{V},\bar{W}$	Unit Vectors in the Radial, Tangential, and Orbit Normal Directions, respectively.
U,V,W	Spacecraft Acceleration Components in the Radial, Tangential, and Normal Directions.

Vectors/Matrices/Functions:

α	Solar Sailcraft Spin Rate about $\bar{b}_3$ axis.
δu	$U_{new} - U_{old}$ = Required Change in Control Vector $\bar{u}$; [4x1].
$[C_i]$	Optimality Vector = λ^T [PC] ; [1x4] ; i = 1,2,3,4.
F^i	State Function [4x1] = $F^i(x, u)$.
$\bar{H}$	Angular Momentum Vector of Solar Sailcraft.
[PC]	Partial Differential of [C] wrt Control Vector U ; [4x4].
[PCI]	Inverse of Matrix [PC] = [4x4].
[PU]	Partial Differential of Function F wrt Control Vector $\bar{u}$; [4x4].
$\bar{U}$	Control Vector = $[\theta, \eta, \zeta, \phi-\psi]$ = $[\theta, \eta, \zeta, PA]$; [1 x 4]
$\bar{S}$	Solar Direction Vector.
$\bar{X}$	State Vector = $[a, e, i, \Omega]$; [1 x 4]

Shadow Components:

R_{limit}	Radius Limit on Extent of Shadow.
R_S	Radius at which Sailcraft enters Shadow = Orbit Radius for Circular orbits.
R_E	Radius of Earth 3444 nmi.
θ_E	Angle between Earth-Sun line to vector at shadow's point of entry . ($\theta_E > 90^\circ$)
θ_S	Shadow Half-Angle (this is the complementary angle to θ_E).
t_{SH}	Time spent in shadow

Miscellaneous Parameters:

di	Constants defined in Equation (2.29); i = 1,2,3.
t	Time.
βi	Constants defined in Equation (2.30); i = 1,2,...,8.
wrt	with respect to

Optimization Parameters:

L°	Lagrangian
H°	Parameter used in Equation (3.4)
J	Performance Index
λ	Lagrange Multiplier used in Equation (3.16)

Abstract

This study addresses the maximization of the circular orbit radius of an earth-orbiting solar sail under free coning motion. The objective is to find the optimal sail settings producing the most change in the semi-major axis per orbit. Angular orientations with respect to sail nutation, precession, mean motion, and its angular momentum control the magnitude of the solar thrust along the sailcraft's velocity direction. A numerical search scheme uses a modified Newton-Rapson iteration method to identify sets of control parameters meeting certain optimality conditions that produce a stationary value in a selected performance index. Such scheme displays its vulnerability to a lack of a good initial guess. Three dimensional perspectives of the small perturbation equation describing the behavior of the change in semi-major axis facilitates the understanding of its cyclic nature and provides an excellent tool for identifying the various locations of possible maxima (and minima) as well as slope-critical area. The perspectives improves the initial guess for implementing maxima search schemes. A test case demonstrates the location of particular points of interest with few search iterations.

I. Introduction

1.1 Solar Sailing

The Concept. Solar Sailing is a method of propelling an object through space by the use of solar radiation pressure. Issac Newton formulated the equations that makes this method realizable and most feasible. Pressure acting on an area equates to some resultant force (force = pressure x area) which, if not acted against, would produce some type of motion in the direction of that force (force = mass x acceleration). One can think of solar sailing as "sailing with the sun". Studies accomplished in the past have shown it to be a most economical method of interplanetary travel. The key to this economy is in the nature of its propulsion system.

The Propulsion System. The basic propulsion element in a solar sail system is a highly specularly reflective mirror-like surface that creates the thrust by reflection of sunlight. The physics of converting photon energy to spacecraft motion is simply this: the solar radiation pressure results from changes in the momentum of incident photons on the sails reflective surface. The higher the reflectivity (i.e., lower absorbtivity) of the sail, the greater would be this momentum transfer. The pressure on the exposed sail surface area constitutes the "thrust" on the vehicle which produces the motion (see Fig. 1.1). This force acts to increase or decrease the sail total energy. This change in energy allows the use of solar sailing as a means of space travel. Controlling the angle of incidence of the incoming solar raditaion (photons) amounts to controlling the direction of this resultant force.

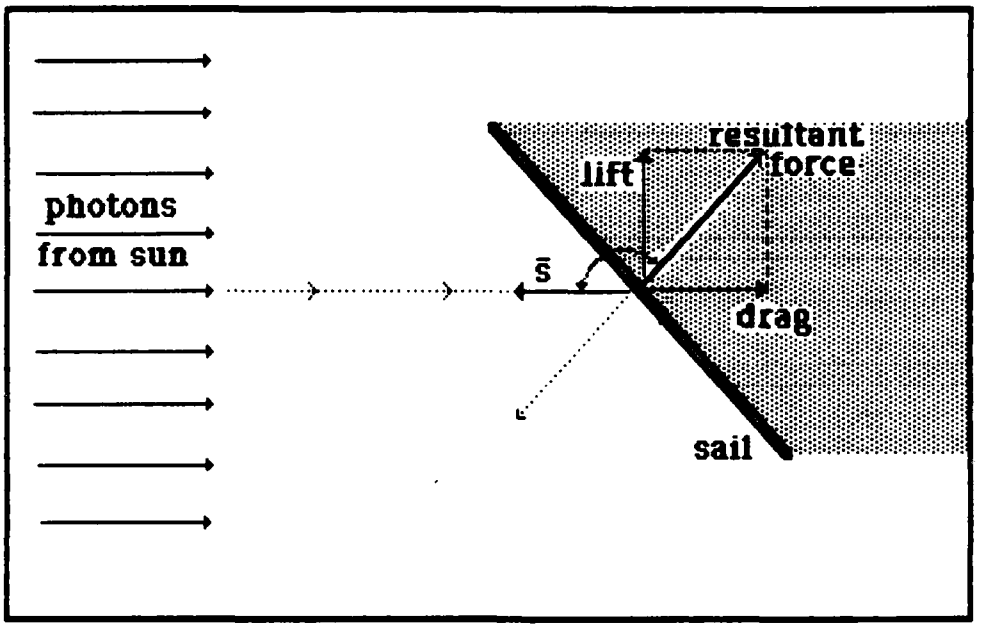


Fig 1.1: Force Vector Diagram

The Controls System. The *only control parameter* in a solar sail propulsion system (with constant sail surface area) is the "setting angle" of the sail i.e., the angle between a unit vector normal to the shady side of the sail and a unit vector in the direction of the sun as shown in Fig. 1.1. The effect of a change in this setting angle is a change in the direction and magnitude of the resultant solar thrust (force). Since the sail changes its orientation with respect to an inertial frame from which the orbital parameters are referenced, this control or setting angle is usually described by a set of orientation parameters.

1.2 Maximizing the Semi-Major Axis

Total Energy Approach. In order to increase the semi-major axis a, the total energy (kinetic plus potential) of the sailcraft must be increased from the energy of the original orbit to some higher energy level. The rate of energy increase is equal to the rate that work is done on the sailcraft by the thrust produced by its propulsion system (solar sails themselves). It may seem advantageous and optimal to direct the thrust such that the rate of work done on the sailcraft is always at or near a maximum.

This is not necessarily true. For circular orbits, maximizing the semi-major axis is essentially maximizing the orbital radius. For small perturbations, this assumption is valid. Maximizing the rate of energy increase is realized for this sailcraft when the thrust component in the velocity vector is maximized. Analyses have shown that the orientation of the sailcraft's orbit with respect to the earth-sun line determines the maximum posible rate of this energy increase. [Ref: 4] All strategies to increase the semi-major axis are indeed energy increase strategies.

1.3 Problem Statement

A Solar sail being in a photon rich enviroment, inevitably experience a perturbational force which affect the orbital characteristics. These effectss are indeed dependent upon the sails orientation with respect to the solar radiation source. To effectively use the resulting thrust created by this radiation pressure to provide changes in the orbital parmeters, the orientation must be known. Random orientation will result in random orbital parameter changes. For space travel, interest lies in the maximum changes in orbital parameter the semi-major axis. Certain combination of solar sail setting angles can provide the best change for a given cost parameter .... be it travel time or the number of revolutions to acquire a desired change.

1.4 Objective and Scope.

The objective of this study is to determine what set of control angles would provide the optimal change in the semi-major axis of an earth-orbiting solar sailcraft. The scope of this effort is to investigate the nature the perturbation of the semi-major axis and determine, if any, the set of control parameters that produce the maximum change in this orbital parameter in one revolution.

1.5. Solution Methodology.

Technique. This effort incorporates the deserption of semi-major axis perturbation function and the determination of conditions that maximize it. The technique of dynamical systems optimization is employed as presented by Bryson and Ho [Ref: 24]

Tools. This approach requires the identification of the perturbation function and its corresponding surface followed by the construction of a computer program to evaluate the sensitivity of the orbital parameters to small perturbational forces and to search for a set of control angles satisfying a given performance index. This index is established as the "maximum change in semi-major axis per orbit".

1.6 Summary of Report.

Section II provides some background information into the previous studies on solar sailing and related topics. Their results are summarized. A description of the solar sail model used in the study along with the basic and simplifying assumptions that led to development of the solar sail coning phenomenon follows. A short rationale for the neglect of any shadowing effects is presented. The general perturbation equation derived by an earlier researcher is examined for a special resonance case and is evaluated for three distinct sailcraft motions: spinning, coning, and tumbling.

Section III discusses the use of surface representations of the perturbation function Δa as a method to identify the regions of maxima and minima. The dynamic system optimization that follows it bases its region-of-search on these surface representations. The approach to the solution via the gradient method is presented.

Section IV provides an insight into the nature of the perturbation equation by graphically portraying the surface of the perturbation function. These surfaces represent the function's behavior from a three-dimensional perspective at critical setting angles and provides the best initial guess for the search scheme.

Section V presents the conclusions drawn from the entire effort. It includes the lessons learned and suggestions for further study.

The Appendices provide a detailed look at the eclipsing (shadowing) phenomenon(Appendix A) along with a rationale for employing the numerical technique in lieu of the analytical (Appendix B). The Fortran Programs used to generate the numerous surfaces and to determine the optimum control settings are included in Appendix C and D, respectively. Appendix E provides a sample output of a test case to show the search mechanism using the Stirling approximation technique.

II. Background

2.1 Previous Efforts

There have been several studies in the past that have placed interest in the solar sailing concepts and in the solar radiation induced orbital perturbation of space structures. Studies in the solar sailing concept tend to deal with space travel and the determination of control laws to provide optimum changes in orbital parameters such as semi-major axis and inclination to attain escape from a planet's gravitational field. Studies in the orbital perturbation due to solar radiation have generally employed numerical and expansions techniques to determine orbital motion due to the "formidability" of multi-first order, nonlinear coupled differential equations.

Tsander. [Ref: 18] , is considered the "father" of solar sailing. His serious investigations of the solar sail problem for spaceflight demonstrated, in principle, the feasibility of making interplanetary flights with the aid of solar pressure.

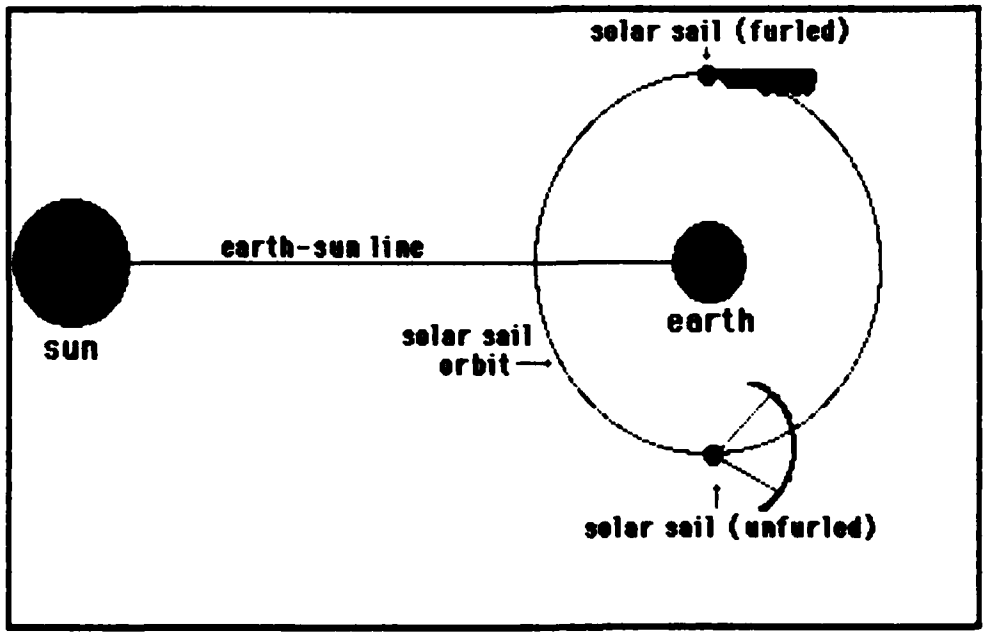


Fig 2.A: Furling/Unfurling Method

[Ref: 5]

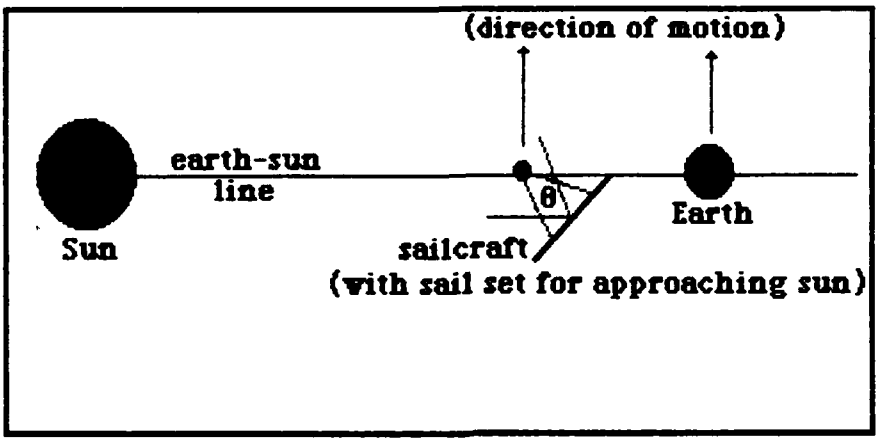


Fig. 2.1B: Optimum Angle Orientation

[Ref: 5]

Garwin. [Ref: 5], whose work in establishing the concept of solar sailing as a practical means for space vehicle propulsion, considers the "furling and unfurling technique to increase the altitude of a sailcraft and escape the earth's gravitational field. Figure 2.1A shows this method. The objective is to maximize change in semi-major axis as the sailcraft travels away from the sun and to minimize the change as it approaches the sun is the apparent rationale for this approach. This study states that an optimum tilt angle is $\theta \approx 35^{\circ}$ for maximizing the semi-major axis. From figure 2.1B this angle will result in the largest thrust along the velocity vector (i.e., along the direction of motion).

Fimple. [Ref: 4], analyzes the use of solar sails for an earth escape trajectory from a circular orbit using the maximum time rating energy increase approach. Trajectory analyses for two different orbit geometries (see Figure 2.2) reveals that orbits parallel to the solar radiation (Orbit A) results in a radical orbit eccentricity change. Orbits normal to the solar radiation (Orbit B) result in a slowly changing orbit eccentricity or quasi-circular trajectories. Hence, Orbit B enhances the change in orbit semi-major axis; launching into this orbit is, however, a difficulty.

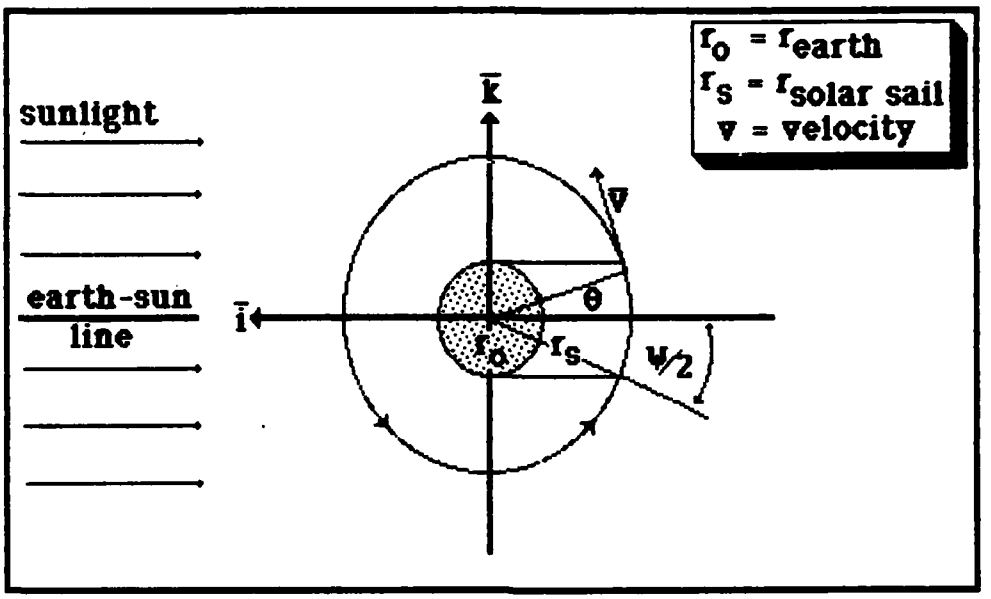


Fig 2.2: Orbit A Geometry

[Ref: 4]

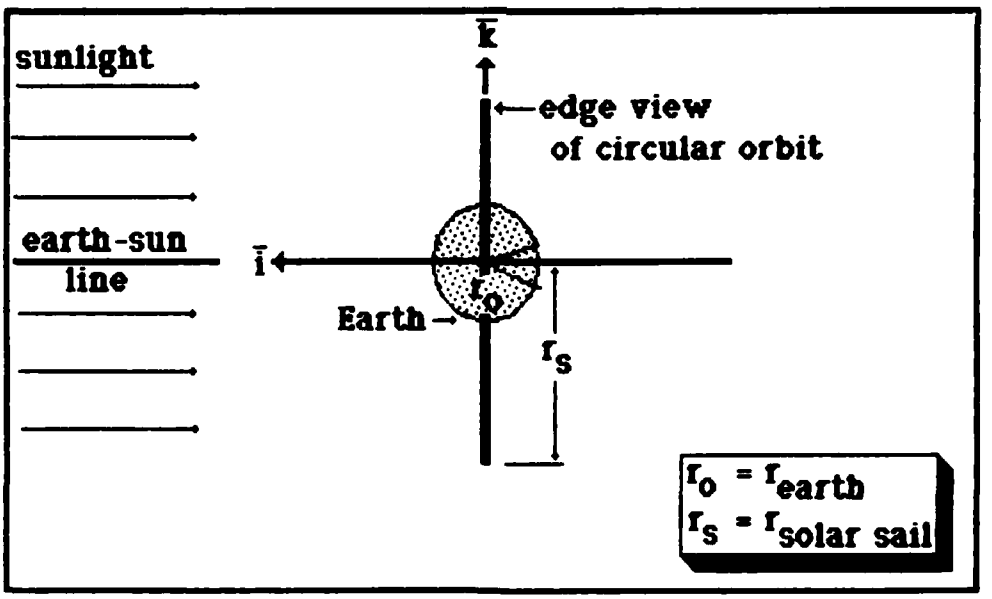


Fig 2.2B: Orbit B Geometry

[Ref: 4]

Sands. [Ref: 15] , studied the similar use of solar sails for escaping the gravitational infuluence of a given planet employing the total energy change approach. Results show that placing the sailcraft in an initially circular orbit spinning at a rate ω - 1/2 period constitutes an inefficent escape maneuver. An alternative suggested was an eliptical orbit with energy level as the original starting orbit.

Cotter. [Ref: 27] , introduces a very important number called the "lightness" of the sail. This lightness is the ratio of the maximum solar radiation pressure force to the solar gravity force on a given sailcraft. Since these two forces are functions of the inverse square law in which its value varies inversely as the square of the distance from the sun, this ratio independent of this distance from the sun. This makes this ratio uniquely a directly proportional measure of the inertia of the sailcraft (i.e., the larger the ratio, the greater the acceleration of the sailcraft).

This study discusses the feasibility of solar sailing and demonstrates it application for interplanetary travel. Travel limitations are in time and temperature. The temperature limitation (operating temperature envelope of the sailcraft) dictates a prudent use of near-sun trajectories; the time limitations restrict planetary travel to the nearer portions of the solar system. The advantage of having unlimited "fuel" supply makes this solar sailing concept most promising.

London. [Ref: 12] . provides some insight into the motion of a solar sailcraft with constant sail settings. The study employs the logarithmic spiral trajectories as a solution and concludes that it (solution) is not optimum. It encourages the use of other trajectories for more efficient sail utilization.

Tsu. [Ref: 19] , evaluates the neccessity and importance of low mass to area ratios for solar sail design along with the travel time and optimization of the sail tilt angle θ for interplanetary trips with minimum time as a constraint. The optimum sail setting for such application is found to be dependent upon the acceleration α of the sailcraft (due to radiation pressure acting on the sail area) and the sun's gavitational acceleration at earth's orbit.

The time of travel between radii r_0 and r_f is

$$t = 1/3 \left\{ \frac{(r_0^{3/2} - r_f^{3/2})}{r_0 \alpha^{1/2}} \right\} \cdot \left\{ \frac{(a_0/\alpha - \cos^3\theta)^{1/2}}{\sin\theta \cos^2\theta} \right\} \qquad (2.1.a)$$

where r_0 - initial orbit radius,
r_f - final orbit radius,
a_0 - sun's gravitational acceleration,
α - sailcraft acceleration due to solar pressure,
θ - angle of incidence.

For shortest time,

$$\frac{d}{d\theta} \left\{ \frac{(a_0/\alpha - \cos^3\theta)^{1/2}}{\sin\theta \cos^2\theta} \right\} = 0 \qquad (2.1.b)$$

from which the optimum tilt angle θ_{opt} can be found. Tsu plots this out and finds that for $a_0 = 0$, the optimum tilt angle θ_{opt} is approximately 35^0. This is the same result found by Garwin [Ref: 5].

Van der Ha and Modi. [Ref: 21A] , have done the most extensive analytical studies of orbital perturbation caused by solar radiation. This study considers a more realistic solar radiation model (than the model presented in section 2.2) and analyze three special orientation cases with a) fixed angle with respect (wrt) to the solar radiation source, b) fixed angle wrt an inertial frame and c) a general fixed orientation wrt the earth.

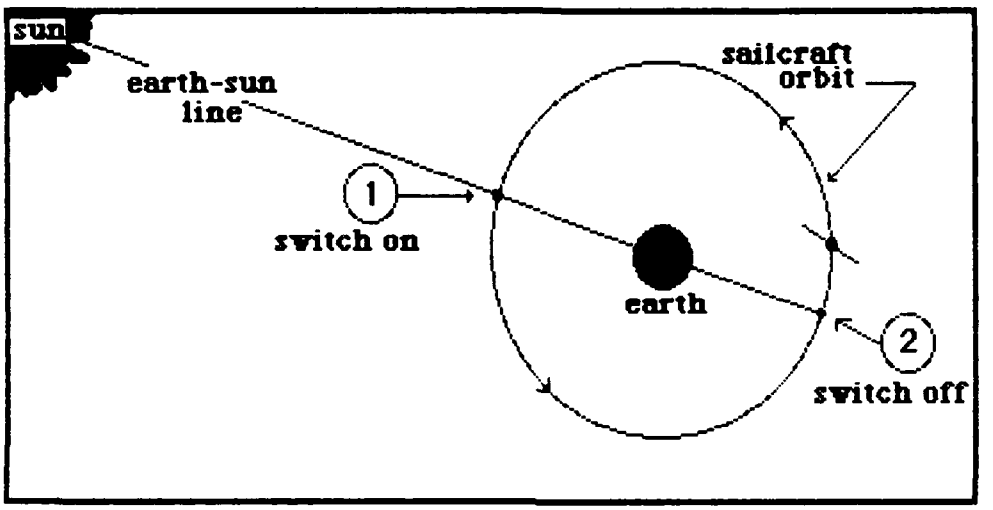


Fig. 2.3: Switching Points Configuration

[Ref: 21A]

In another study [Ref: 21C], these same authors employed on/off switching strategies to increase the semi-major axis. Such strategies involve instantaneous switching controls which in effect turns the solar pressure force off during certain parts of the orbit. The strategy calls for "switch-on" when the sailcraft is (roughly) near the earth-sun line (point 1) and "switch-off" at (point 2) as shown in Figure 2.3 (circular orbit shown).

Although instantaneous on/off switching may be technically impractical, it is theoretically effective in the sense that the rate of change of the total energy is always positive during the on-phase since the component of the perturbing force along the instantaneous velocity vector is positive.

Jenkins. [Ref: 10] . is the only individual to date who has explored the behavior of the coning solar sail. The coning motion of a given sailcraft greatly affects the magnitude and direction of solar radiation pressure force. His study generated the perturbation algorithms that determines the changes in orbital parameters (a, e, i, Ω) as a function of specified sail setting angles which relate the sailcraft orientaition with respect to an inertial frame.

2.2 The Coning Solar Sail.

Solar Sail Model. In general, a solar sailcraft must meet one critical criterion before it can literally "sail" anywhere. The mass-to-area ratio is this critical parameter that must be established. It is the ratio of the total solar sail area to the total mass of the sailcraft (sail + structure + electronics + payload). Studies have shown that the mass/area must be very low. This is quite apparent from the equation for acceleration due to solar measure. The solar acceleration due to solar pressure is expressed as

$$\alpha_{\text{solar pressure}} = p_0 \, A/M \qquad (2.2)$$

where p_0 - solar radiation pressure at earth's orbit radius.

A - Area of Solar sail and M - total mass of solar sailcraft.

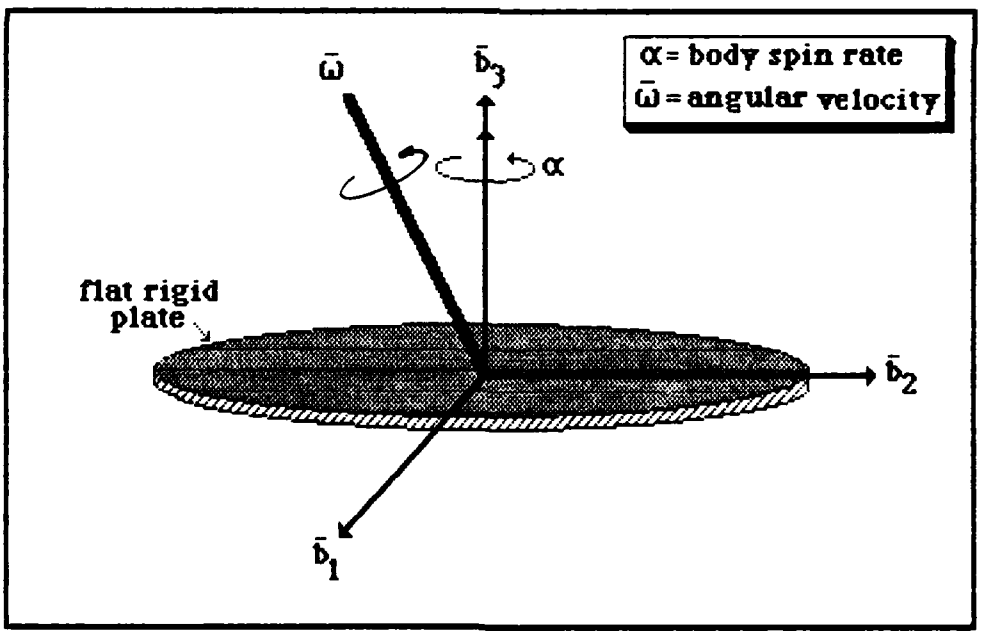


Fig 2.4: Solar Sail Model

[Ref: 10]

Typical values for p_0 is about $.9 \times 10^{-4}$ dyne/cm^2. For any appreciable amount of acceleration for space travel, A/M must be high (or its reciprocal M/A be very low). This essentially requires that sail area A be much larger with respect to the sailcraft mass. [Ref: 19]

The solar sail model used in this effort is modeled as a perfectly, specularly reflective (both sides) thin, flat rigid plate as shown above in Figure 2.4. Recall, a highly specular surface means that all incident photons are reflected. Hence, the solar sail is assumed to have a reflectivity of 1.0 which implies that all incident radiation is converted to thrust. Having both sides with the same sail characteristics means that all surfaces exposed to direct solar radiation will be contributing to the overall thrust of the vehicle. The A/M ratio for this model is assumed large enough to make its design feasible for solar sailing.

Basic Assumptions. The basic assumptions made in the course of this effort and in the development of the perturbation solution which follows are enumerated below categorically:

A. For satellite motion, it is assumed that,....

1a. the solar sailcraft is initially injected into a high altitude circular orbit;

2a. the gravitational field is central;

3a. perturbations to the central force gravity field due to oblateness and the gravitational effect of other celestial bodies (e.g., moon, sun) are neglected; also neglected are any magnetic distorbances.

4a. the solar sail motion is torque-free; therefore, the angular momentum $\bar{H}$ is conserved.

5a. the angular velocity vector $\bar{\omega}$ is not necessarily colinear with the

angular momentum vector $\bar{H}$;

6a. for any orientation, $d\theta/dt = 0 \rightarrow$ nutation angle θ = constant;

7a. for any orientation, $d\nu/dt = 0 \rightarrow$ precession rate ν = constant;

8a. orbit plane is considered inertial;

9a. perturbational changes are small; and

10a. the body spin rate α is large enough for spin stabilization.

B. For the solar sail it is assumed that....

1b. the spinning solar sail is an axi-symmetric rigid body whose sail surface does not deform under loads;

2b. the solar sail surface characteristic are essentially homogenous and time invariant;

3b. the solar sail is perfectly specular reflective on both sides; and

4b. the mass/area ratio $\ll 1.0$.

C. For solar radiation, it is assumed that....

1c. reflection of earth surface is neglected; the sun is the only source of radiation; and

2c. no shadowing or eclipsing of sailcraft occurs. This is due to the high altitude and the inclination of the orbit in case.

Torque-Free Motion and the Coning Phenomenon. This section considers the torque motion of the solar model which is axis-symmetric and has a principal Moment of Inertia [A' A' C'] with A' along the symmetric body axes and C' along the $\bar{b}_3$ axis perpendicular to the plane of the sail. The general equation of torque-free motion is

$$\Sigma \bar{M}_{EXT} = d\bar{H}/dt = 0 \qquad (2.3)$$

since (by definition) Torque-Free ⇒ moment $\bar{M}$ equals zero, $\bar{H}$ must be constant in both direction and magnitude; i.e.,

$$\bar{H} = \bar{H}_0 \qquad (2.4)$$

where $\bar{H}_0$ is the initial angular momentum of the sailcraft about its center of mass (which is assumed to coincide with its geometric center). The direction of $\bar{H}$ is fixed in inertial space and can, therefore, can be arbitrarily referenced to an inertial frame. Figures 2.5A & 2.5B depict this solar sailcraft with respect to three different reference frames: [Ref: 2]

a) Body Fixed Frame: $(\bar{b}_1\ \bar{b}_2\ \bar{b}_3)$

b) Orbital (inertial) Frame: $(\bar{i}\ \bar{j}\ \bar{k})$

c) Body-Centric (inertial) Frame: $(\bar{I}\ \bar{J}\ \bar{K})$

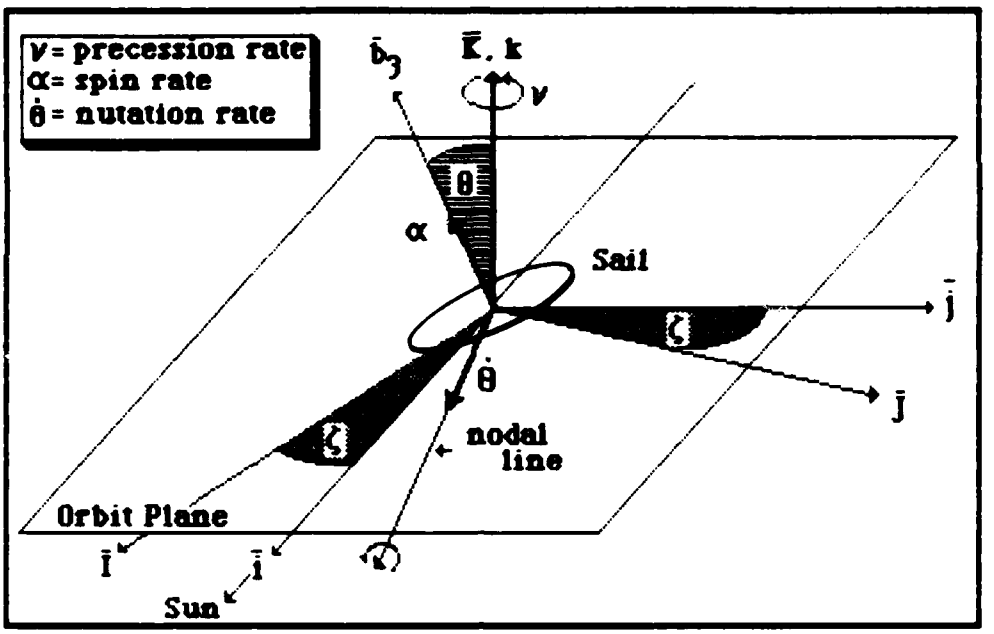


Fig. 2.5A: Reference Frames

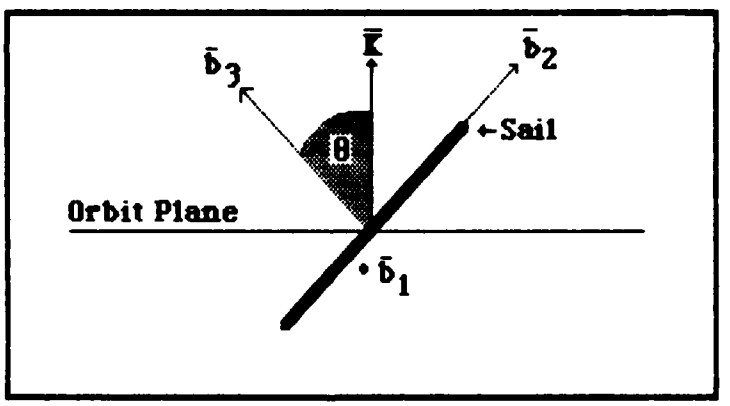


Fig. 2.5B: Coning Angle (θ) Orientation

Using the same angle notations as found in [Ref: 10], $\bar{H}_0$ can be expressed in terms of Euler angles as follows:

$$\bar{H}_0 = H_0 \sin\theta \sin\alpha t \; \bar{b}_1 + H_0 \sin\theta \cos\alpha t \; \bar{b}_2 + H_0 \cos\theta \; \bar{b}_3 \qquad (2.5a)$$

However, for principal axis, this $\bar{H}_o$ equates (by definition) to the following expression:

$$\bar{H}_o = A' \omega_1 \bar{b}_1 + A' \omega_2 \bar{b}_2 + C' \omega_3 \bar{b}_3 \qquad \text{[Ref: 9]} \qquad (2.5b)$$

where [A' A' C'] are the Principal Moments of Inertia. In matrix notation, this can be expressed readily as follows:

$$H_o = \begin{bmatrix} H_o \sin\theta \sin\alpha t \\ H_o \sin\theta \cos\alpha t \\ H_o \cos\theta \end{bmatrix} \quad \begin{Bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \end{Bmatrix} = \begin{bmatrix} A' \omega_{b1} \\ A' \omega_{b2} \\ C' \omega_{b3} \end{bmatrix} \quad \begin{Bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \end{Bmatrix} \qquad (2.6a)\ (2.6b)\ (2.6c)$$

Equating term per term and solving for the angular velocity about each body axis, ...

$$\omega_{b1} = \frac{H_o \sin\theta \sin\alpha t}{A'} \rightarrow \dot{\omega}_{b1} = \frac{d}{dt}\left\{\frac{H_o \sin\theta \sin\alpha t}{A'}\right\} \qquad (2.7a)$$

$$\omega_{b2} = \frac{H_o \sin\theta \sin\alpha t}{A'} \rightarrow \dot{\omega}_{b2} = \frac{d}{dt}\left\{\frac{H_o \sin\theta \cos\alpha t}{A'}\right\} \qquad (2.7b)$$

$$\omega_{b3} = \frac{H_o \cos\theta}{C'} \rightarrow \dot{\omega}_{b3} = \frac{d}{dt}\left\{\frac{H_o \cos\theta}{C'}\right\} = 0 \qquad (2.7c)$$

The Euler equations for dynamical motion (as applied to this sailcraft) are given as follows: [Ref: 11]:

$$\Sigma M_{b1} = A' \dot{\omega}_{b1} - (A' - C') \omega_{b2}\omega_{b3} = 0 \qquad (2.8a)$$

$$\Sigma M_{b2} = A' \dot{\omega}_{b2} - (C' - A') \omega_{b3}\omega_{b1} = 0 \qquad (2.8b)$$

$$\Sigma M_{b3} = C' \dot{\omega}_{b3} = 0 \qquad (2.8c)$$

Applying the respective values for the time derivatives of the angular velocities, the results are as follows.

$$A' \frac{d}{dt}\left[\frac{H_0 \sin\theta \sin\alpha t}{A'}\right] + \frac{(C' - A')}{A'C'} H_0^2 \sin\theta \cos\theta \sin\alpha t = 0 \quad (2.9a)$$

$$A' \frac{d}{dt}\left[\frac{H_0 \sin\theta \cos\alpha t}{A'}\right] + \frac{(A' - C')}{A'C'} H_0^2 \sin\theta \cos\theta \sin\alpha t = 0 \quad (2.9b)$$

$$A' \frac{d}{dt}\left[\frac{H_0 \cos\theta}{C'}\right] = 0 \quad (2.9c)$$

From Equation (2.9c), the derived conclusion is that $H_0(\cos\theta)/C'$ - Constant; hence,

$$\theta = \theta_0 = \text{constant} \quad (2.10)$$

since H_0 and C'are established as constants. Hence the nutation angle or the coning angle remains fixed.

Using this fact in Equation (2.9b) and carrying out the differentiation, the result is as follows: The magnitude of the angular velocity is.....

$$\alpha = \frac{(A' - C')}{A'C'} H_0 \cos\theta_0 \quad (2.11)$$

from which is known that (recall, the Moment of Inertia terms and H_0 are constant)...

$$\alpha \equiv \text{the body angular velocity - constant.} \quad (2.12)$$

The precession is found from the components of the angular velocity about body axis b_3. This happens to be (as reference from Figure 2.5A)

$$\omega_{b3} = \alpha + \nu \cos\theta = H_0 \cos\theta/C' \quad (2.13)$$

Solving for ν (with α as expressed above), the result is

$$\bar{\nu} = \bar{H}_0 / A = \text{constant} \tag{2.14}$$

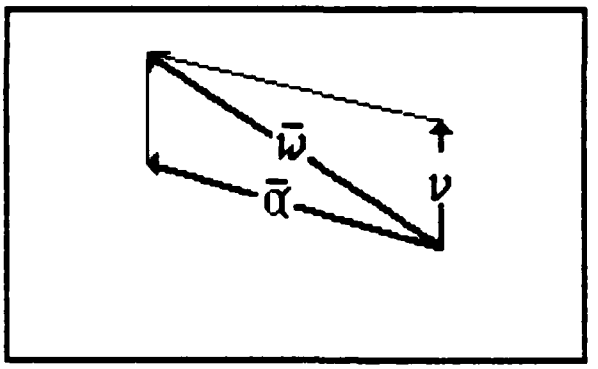


Fig. 2.6: Velocity Vectors

Solving for the total angular velocity $\bar{\omega}$ is the vector sum of $\bar{\alpha}$ and $\bar{\nu}$ (see Figure 2.6).

$$\bar{\omega} = \bar{\alpha} + \bar{\nu} \tag{2.15}$$

and would be at a set angle ε from the $\bar{K}$ direction. With $\bar{\nu}$ (precession rate) fixed along $\bar{K}$ direction and $\bar{\alpha}$ (body spin) fixed along body $\bar{b}_3$ direction, this phenomenon displays the vector $\bar{\omega}$ as "sweeping out" a cone. The resultant motion of this axis-symmetric rigid body in which the angular velocity vector $\bar{\omega}$ is not colinear with the angular momentum vector $\bar{H}$ is known as "coning". This behavior is shown in Figure 2.7.

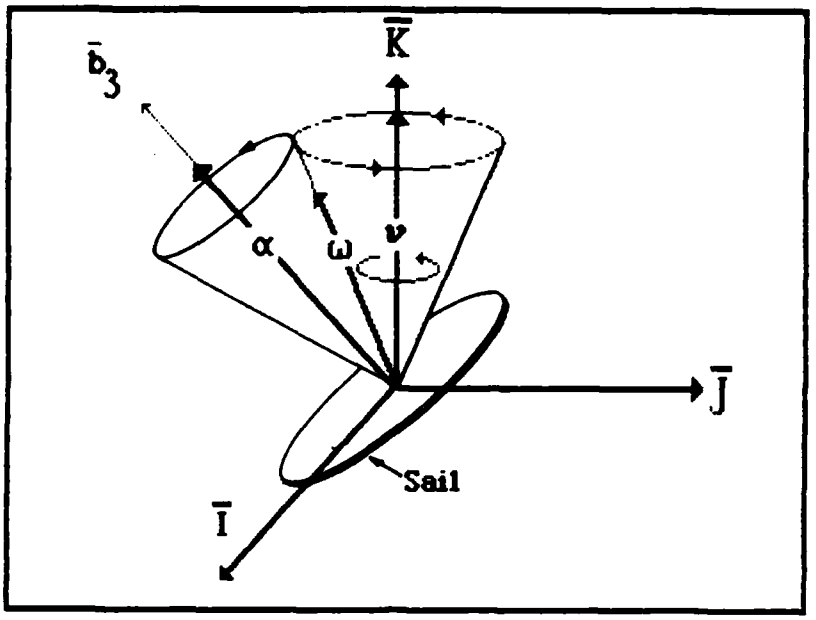


Fig. 2.7: Coning Motion

[Ref: 2.10]

The coning angle (as referred to in this text) is θ. It, too, displays the "sweeping" cone phenomenon. This is easily seen with a projection of the body $\bar{b}_3$ axis on the IJ plane. This is accomplished in Figure 2.8 . Line $\overline{OX}$ is this projection and it precesses about the $\bar{K}$ axis at a rate ν, the precession rate.

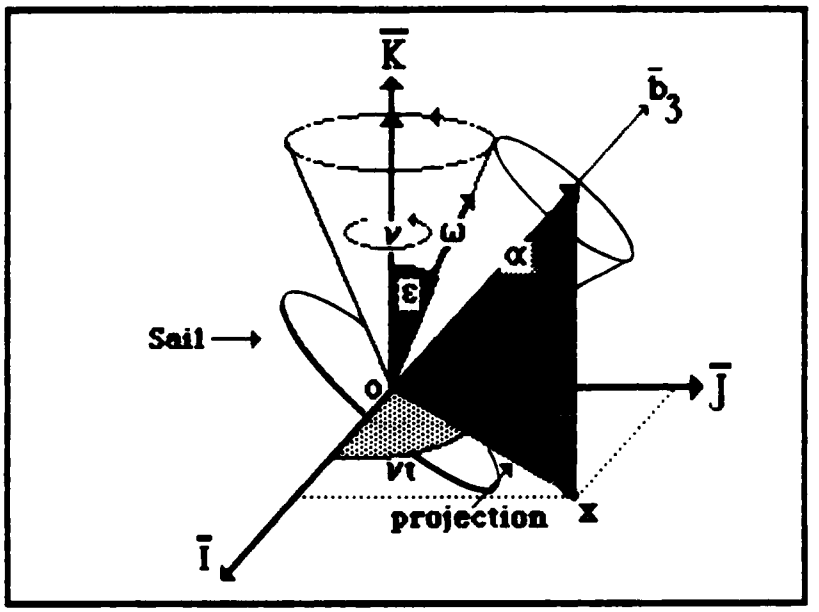


Fig. 2.8: Precession

[Ref: 2.9]

The Orientation Angles (θ, η, ζ, ϕ-ψ). The coning angle θ (as discussed previously) is the angle between the angular momentem vector $\bar{H}$ and the Solar Sail body b_3 axis [Ref: Fig. 2.5B]. The Angular momentum vector $\bar{H}$ is referenced to the orbital plane via angles η and ζ as shown in Figure 2.9A below:

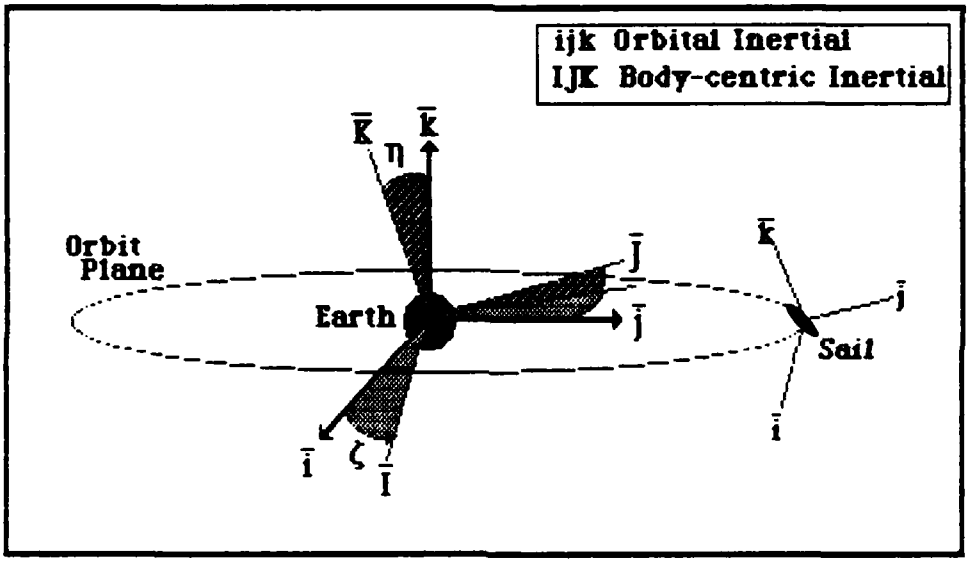


Fig. 2.9A: η and ζ Orientation

[Ref: 10]

The $\bar{I}\bar{J}\bar{K}$ Frame is defined to be sail body-centered with the $\bar{I}$ axis in a plane perpendicular to the orbital plane. This study refers heavily on the difference between two phase angles, ϕ and ψ. This difference, suitably called PA, equals ϕ-ψ i.e.,

$$PA = \phi - \psi \tag{2.16}$$

1. Angle Φ references the solar thrust components in the $\bar{U}\bar{V}\bar{W}$ directions (U ≡ radial, V ≡ tangential, and W ≡ normal) to the Orbital (i j k) Frame. Angle ϕ then fixes the initial position of the sailcraft in its orbit. At $\phi = 0°$, the sailcraft will be initially between the earth and the sun for ectiptic motion or in a plane perpendicular to orbit plane and laying in the earth-sun line.

2. Angle Ψ denotes the phase of the Body ($b_1b_2b_3$) Frame's precession about the angular momentum vector $\bar{H}$ to the Body-Centric (IJK) Frame. These are shown below in their respective coordinates orientation Figures 2.9B and 2.9C.

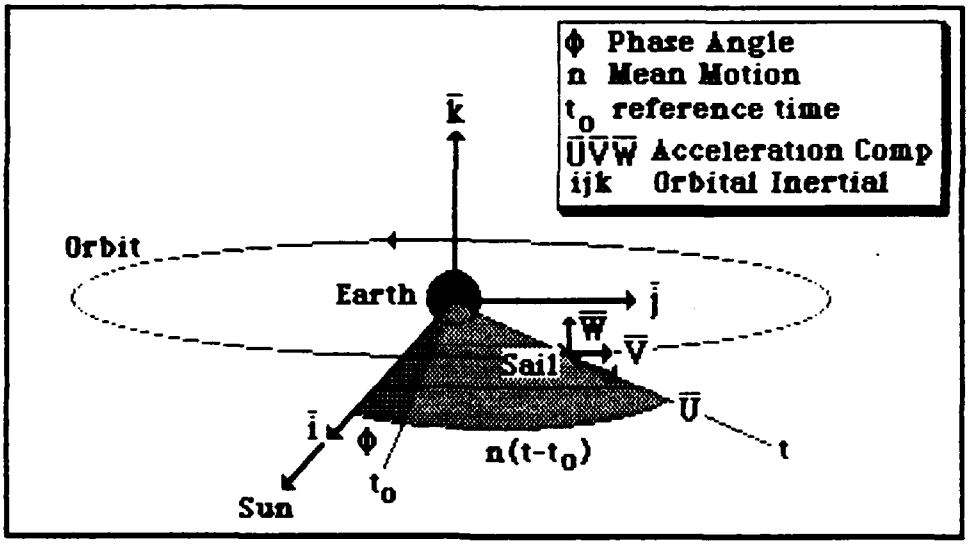


Fig. 2.9B: Angle ϕ Orientation

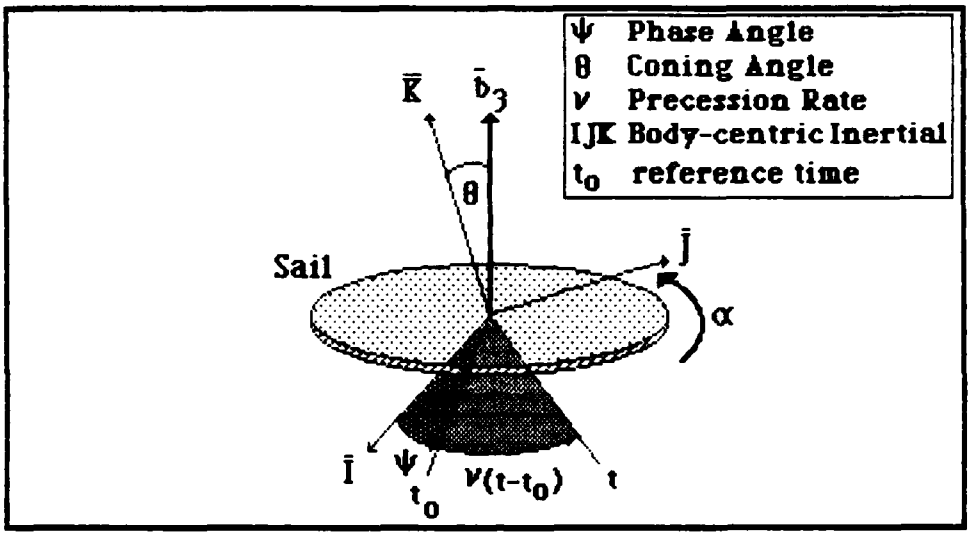


Fig. 2.9C: Angle ψ Orientation

With, $\theta = \eta = \zeta = 0°$, the $(\phi-\psi)$ angle can be best seen for understanding as is done in Figure 2.9D below:

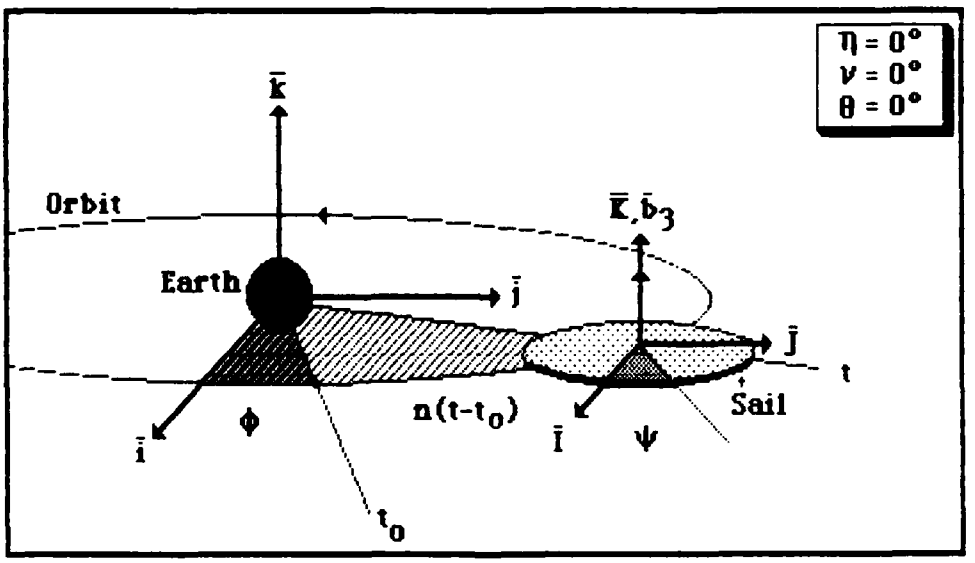


Fig. 2.9D: Angle $(\phi-\psi)$ Orientation

2.3 Shadowing Or Eclipsing Effect.

Eclipsing or shadowing of the solar sail was not considered in the formulation of the perturbation solutions. Investigation of this shadow phenomenon lead to the conclusion that for a given orbit inclination and altitude, shadow effects can be ignored. From a general geometrical perspective, one can safely conclude that shadowing will occur if the orbit lies in the ecliptic (earth-sun) plane. This is theortically true. The interesting question is this: At what orbit inclination and altitude is shadowing a problem? Stoddard (Ref: 25) evalutes this phenomenon and arrives at a set of equations that determine the...

a) minimum inclination before shadowing effects can be considered dominant and b) the altitudes a satellite must be at to avoid shadowing.

Fixler (Ref: 6) circumvents this entire issue by studying solar sailing at orbits perpendicular to the ecliptic plane. This approach surely removes any uncertainties of shadow effects completely. Solar sails, as dependent as they are to the amount of incident radiation for propulsion, can be so placed in an orientation that minimizes this eclipsing phenomenon.

Figure 2.10 below shows the shadow and no-shadow situations. From this, one can state that there does exist some physical limit to the shadowing. Stoddard explains and develops a method to analyze this situation. He presents his arguments using the cylindrical shadow theory and concludes that the critical inclination for a given orbit radius is...

$$i \approx \cos^{-1} [R_e/R_s] \tag{2.17}$$

Stoddard claims that for

$$R_s < R_e \csc i \Rightarrow \text{an eclipse will take place.} \tag{2.18a}$$

$$R_s > R_e \csc i \Rightarrow \text{an eclipse will } \underline{\text{not}} \text{ take place.} \tag{2.18b}$$

His arguments on this issue, along with a determination for the duration of shadowing (if any), are provided in Appendix A. The results are interesting. They show the comparison of a system with and without shadow effects accounted for. The bottom line is this: For low-earth orbit solar sails within the shadow limits, shadowing effects are dominant and must be considered. By increasing the orbit inclination and the altitude, a satellite can circumvent this shadowing. This shadowing phenomenon and its effects are circumvented in this study by employing non-ecliptic planes and orbit altitudes beyond the shadow regime. With this understood, the tasks of analyzing solar sail behavior in a photon rich environment is pursued here.

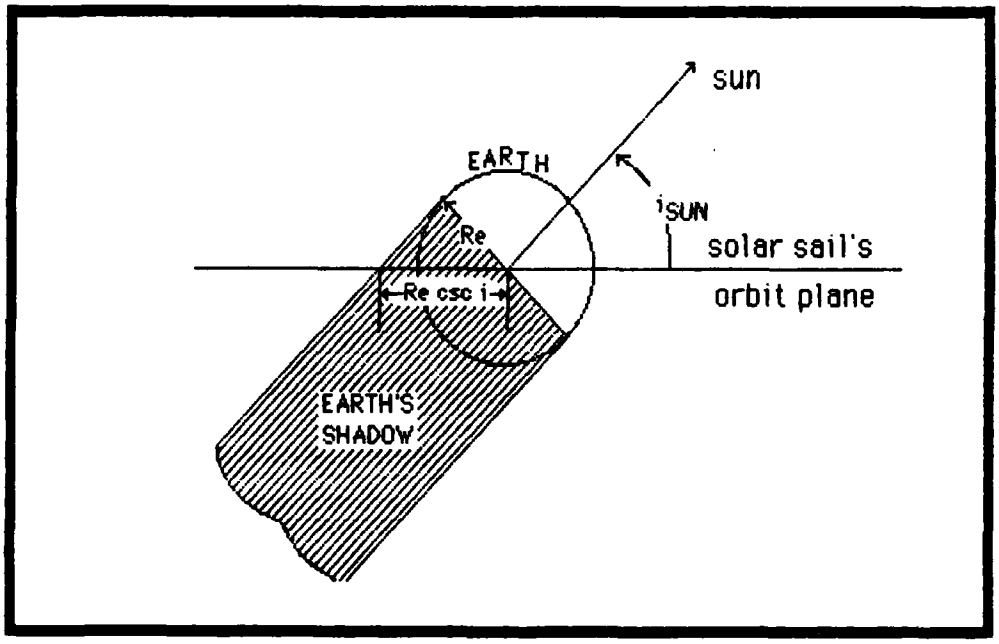


Fig. 2.10: Shadow Geometry

[Ref: 26]

2.4 General Perturbation Equation.

Continuing Work. This study continues the work initiated by Jenkins (Ref: 10) on the study of the "Orbital Motion of Coning Solar Sails" in which he succeeded in developing a set of algorithms for determining the perturbations in certain orbital parameters (a, e, i, Ω) as a function of the solar sail settings. These solar settings are just the orientation angles (θ, η, ζ, ϕ-ψ) of the solar sail with respect to the solar vector $\bar{S}$. Since these setting angles control the behavior of the solar sailcraft, it was of particular interest to determine what angle (s) is/are dominant in producing the greatest changes in the orbital parameter(s) selected. This set of algorithms was derived from the Lagrangian Planetary Equations and are presented below in a more modular fashion than found in [Ref: 10].

Perturbation Equations.

$$\Delta a = \int_0^T (da/dt)\ dt \tag{2.19a}$$

$$\Delta e = \int_0^T (de/dt)\ dt \tag{2.19b}$$

$$\Delta i = \int_0^T (di/dt)\ dt \tag{2.19c}$$

$$\Delta \Omega = \int_0^T (d\Omega/dt)\ dt \tag{2.19d}$$

For circular orbits (e = 0), the Langrangian Planetary Equations reduce to the following:

$$da/dt = \frac{2D(\bar{b}_3 \bullet \bar{S})^2 (\bar{b}_3 \bullet \bar{V})}{n} \tag{2.20a}$$

$$de/dt = \frac{D(\bar{b}_3 \bullet \bar{S})^2 (\bar{b}_3 \bullet \bar{U}) \sin f + 2D[(\bar{b}_3 \bullet \bar{S})^2 (\bar{b}_3 \bullet \bar{V}) \cos f]}{na} \tag{2.20b}$$

$$di/dt = \frac{D(\bar{b}_3 \bullet \bar{S})^2 (b_3 \bullet \bar{W}) \cos f}{na} \tag{2.20c}$$

$$d\Omega/dt = \frac{D(\bar{b}_3 \bullet \bar{S})^2 (\bar{b}_3 \bullet \bar{W}) \sin f}{na \sin i} \tag{2.20d}$$

with D = Sail constant - 3k Ag/m

f - true anomaly

$\bar{b}_3$ - body vector $\bar{b}_3$ expressed in orbital reference frame

$\bullet$ ≡ dot product

$\bar{S}$ - solar vector (direction of sun)

$(\bar{U}\bar{V}\bar{W})$ - (Radial, Tangential, Normal) components of thrust per unit mass.

<u>One-To-One Resonance Perturbation</u>. In the context of this study, the one-to-one resonance occurs when the mean motion of the satellite about its planet

equals its precession rate about it angular momentum vector H. If the integration in Equation 2.19 is taken over one orbital period with the orbital motion (n) equal to its precession rate (ν), the following expressions result:

$$1)\quad \Delta a = a^{1.5}\, D\, TP/\mu \left[da_1 + da_2 + da_3 + da_4 + da_5 + da_6\right] \qquad (2.21)$$

where

$$da_1 = \tfrac{1}{4} D_1^2 \left[(\beta 2+3\beta 4)\cos(\phi-\psi) - (3\beta 1+\beta 5)\sin(\phi-\psi)\right] \qquad (2.22a)$$

$$da_2 = \tfrac{1}{2} D_1 D_2 \left[(\beta 2-\beta 4)\sin(\phi-\psi) + (\beta 5-\beta 1)\cos(\phi-\psi)\right] \qquad (2.22b)$$

$$da_3 = 2 D_1 D_3 \left[\beta 6\cos(\phi-\psi) + \beta 3\sin(\phi-\psi)\right] \qquad (2.22c)$$

$$da_4 = 2 D_2 D_3 \left[\beta 3\cos(\phi-\psi) + \beta 6\sin(\phi-\psi)\right] \qquad (2.22d)$$

$$da_5 = \tfrac{1}{4} D_2 D_2 \left[(3\beta 2+\beta 4)\cos(\phi-\psi) - (\beta 1+3\beta 5)\sin(\phi-\psi)\right] \qquad (2.22e)$$

$$da_6 = D_3^2 \left[(\beta 2+\beta 4)\cos(\phi-\psi) - (\beta 1+\beta 5)\sin(\phi-\psi)\right] \qquad (2.22f)$$

$$2)\quad \Delta e = a^{\frac{1}{2}}\, D\, TP/8\mu \left[de_1 + de_2 + de_3 + de_4 + de_5\right] \qquad (2.23)$$

where

$$de_1 = D_1^2 \left[-\beta 3\sin(2(\phi-\psi)) + \beta 6\,(6+\cos(2(\phi-\psi)))\right] \qquad (2.24a)$$

$$de_2 = 2 D_1 D_2 \left[\beta 3\cos(2(\phi-\psi)) + \beta 6\sin(2(\phi-\psi))\right] \qquad (2.24b)$$

$$de_3 = 2 D_1 D_3 \left[(\beta 2+\beta 4) \cos(2(\phi-\psi))-(\beta 1+\beta 5) \sin(2(\phi-\psi)) + 6 \beta 4\right] \quad (2.24c)$$

$$de_4 = 2 D_2 D_3 \left[(\beta 2+\beta 4) \sin(2(\phi-\psi))+(\beta 1+\beta 5) \cos(2(\phi-\psi))-6\beta 5\right] \quad (2.24d)$$

$$de_5 = 12 D_3^{\,2} \beta 6 \quad (2.24e)$$

$$3) \quad \Delta i = a^{1/2} DTP/2\mu \left[di_1 + di_2 \right] \quad (2.25)$$

where

$$di_1 = 2\beta 8 \; D_3 \left[D_1 \cos(\phi-\psi) + D_2 \sin(\phi-\psi)\right] \quad (2.26a)$$

$$di_2 = \beta 7 \left[(D_1^{\,2}+.75 (D_2)^2+ D_3^{\,2}) \sin(\phi-\psi) +.5 \; D_1 D_2 \cos(\phi-\psi) \right] \quad (2.26b)$$

$$4) \quad \Delta\Omega = a^{1/2} DTP /(2\mu \sin i) \left[d\Omega_1 + d\Omega_2\right] \quad (2.27)$$

where

$$d\Omega_1 = \beta_7\left[(.25 D_1^{\,2} + .75 D_2^{\,2} + D_3^{\,2}] \cos(\phi - \psi) -.5 D_1 D_2 \sin(\phi - \psi)\right] \quad (2.28a)$$

$$d\Omega_2 = 2\beta_8 \; D_3 \left[D_1 \sin(\phi-\psi) - D_2 \cos(\phi - \psi)\right]; \quad (2.28b)$$

The other parameters are as follows:

$$D_1 = \sin\theta \cos\zeta \cos i \quad (2.29a)$$

$$D_2 = \sin\theta \sin\zeta \cos\eta \cos i + \sin\theta \sin\eta \sin i \quad (2.29b)$$

$$D_3 = \cos\theta \sin\zeta \sin\eta \cos i + \cos\theta \cos\eta \sin i \qquad (2.29c)$$

and

$$\beta 1 = \sin\theta \cos\zeta \qquad (2.30a)$$

$$\beta 2 = \sin\theta \sin\zeta \cos\eta \qquad (2.30b)$$

$$\beta 3 = \cos\theta \sin\zeta \sin\eta \qquad (2.30c)$$

$$\beta 4 = \sin\theta \sin\zeta \qquad (2.30d)$$

$$\beta 5 = \sin\theta \cos\eta \qquad (2.30e)$$

$$\beta 6 = \cos\theta \cos\zeta \sin\eta \qquad (2.30f)$$

$$\beta 7 = \sin\theta \sin\eta \qquad (2.30g)$$

$$\beta 8 = \cos\theta \cos\eta \qquad (2.30h)$$

It is quite evident from the above equations how the coning angle θ, along with the orientation angles (η ,ζ ,ϕ - ψ) can <u>control</u> the changes in each orbital parameter a,e,i, and Ω. Judicious choice of these angles can lead to maximum changes in orbital state parameters .

Special Cases. Jenkins [Ref:10] derives certain conclusions with a solar sail system which is in one-to-one resonance and whose motion is in the ecliptic plane. That analysis shows that there exist unique certain sail setting providing maximum changes in the semi-major axis in each of the three cases below:

a) spinning

b) tumbling

c) coning

By singularly varying the sail setting angles in the perturbation equations (2.21 - 2.30) the condition for maximum change in semi-major axis can be graphically shown. This fact is later shown from a three dimensional perspective.

Spinning Case. The solar sail assumes a strictly spinning orientation when the coning angle θ equals 90 degrees. There is no "coning" behavior. This configuration is shown in Figure 2.11. For motion in the ecliptic plane ($i = 0$) with the angular momentum vector $\bar{H}$ aligned perpendicular to the orbital plane ($\eta = \zeta = 0^\circ$), this "spinning" orientation affords the most semi-major axis change. As will be shown later, this is also true for a certain range of orbital inclinations ($0^\circ \leq i \leq 45^\circ$). At inclinations greater than 45°, the spinning orientation actually results in a decrease in semi-major axis.

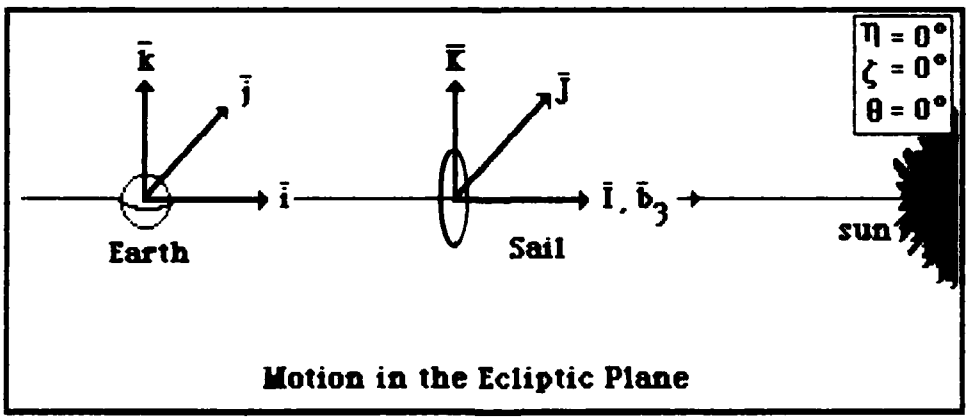


Fig. 2.11: Spinning Case

Tumbling Case. This is the case where the angular momentum vector $\bar{H}$ is parallel to i or j orbital plane reference components. With $\bar{H}$ parallel to the i orbital axis, the solar sail experiences no change in any of the orbital parameters. This is the case when the solar sail is parallel to the solar radiation and hence no resultant thrust component is produced. With $\bar{H}$ parallel to the direction vector, the changes are in the radial and normal direction. This results in changes in the semi-major axis and in the inclination angles which are coupled via the $\phi - \psi$ phase angle. The strong implication is that conditions favoring a change in a will also result is an accompanying change in inclination (which can be unwanted).

Coning Case. Coning occurs, when the sailcraft's axis of symmetry b_3 is not colinear with the angular momentum vector $\bar{H}$. Jenkins investigated this coning motion with $\theta = 45^\circ$ and discovered the importance of the $(\phi - \psi)$ parameter in controlling the magnitude of Δa, and that there exist trade off possibility between Δa and Δi, the resultant change is inclination. The nature of this coning action is further investigated in this study.

III. Discussion

3.1 Dynamic System Optimization.

Single-Stage System. Bryson and Ho (Ref: 24) describe the single-stage transition from some initial state x(0) to a new state x(1) via some choice of control vector u(0) and a given Operating Function F° mathematically as.....

$$x(1) = F^{\circ}[\, x(0), u(0)] \tag{3.1}$$

and schematically as ....

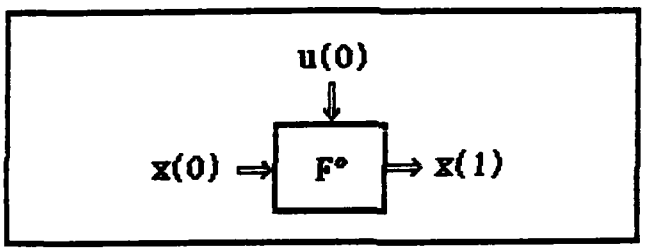


Fig. 3.2: Flow Chart for a Single-Stage System

[Ref: 24]

with a performance index of the form

$$J = \phi\,[x(1)] = L^{\circ}[\, x(0), u(0)] \tag{3.2}$$

where

x(0) - known state at initial time t_0. (n-dimensional)

u(0) - control vector (θ , η , ζ , ϕ - ψ) (m-dimensional)

x(1) - State at some future time t_i

$L^{\circ}[\ ]$ - Lagrangian of the initial state

The objective of an optimization problem leads to the maximization of this performance index J . Bryson and Ho present an adjoined performance index $\bar{J}$ by adding $(F^{\circ}[x(0), u(0)] - x(1) = 0))$ to Equation (3.2).

$$\bar{J} = \phi [\, a(1)] + L^{\circ} \; [x(0), u(0)] = \lambda^{T}(1) \left\{ F^{\circ}[x(0),u(0)] - x(1) \right\} \qquad (3.3)$$

with constant Langrangian multiplier $\lambda^{T}(1)$

Introducing $$H^{\circ} = L^{\circ}[x(0), u(0)] + \lambda^{T}(1) F^{\circ}[x(0), u(0)] \qquad (3.4)$$

into equation (3.3) results in.....

$$\bar{J} = \phi [x(1) + H^{\circ}[x(0), u(0), \lambda(1)] - \lambda^{T} x(1)] \qquad (3.5)$$

The total derivative of J is expressed as

$$d\bar{J} = \frac{\left(\partial\phi - \lambda^{T}(1) \right)}{\partial x(1)} dx(1) + \frac{\partial H^{\circ}}{\partial x(0)} dx(0) + \frac{\partial H^{\circ}}{\partial u(0)} du(0) \qquad (3.6)$$

With the judicious choice of $\lambda^{T}(1) = \partial\phi/\partial x(1)$. Equation (3.6) simplifies to the following:

$$d\bar{J} = \frac{\partial H^{\circ}}{\partial x(0)} dx(0) + \frac{\partial H^{\circ}}{\partial u(0)} du(0) \qquad (3.7)$$

where

$\partial H^*/\partial x(0) \equiv$ gradient of $\bar{J}$ wrt x(0), holding u(0) constant and satisfying Equation (3.1); and

$\partial H^*/\partial u(0) \equiv$ gradient of $\bar{J}$ wrt u(0), holding x(0) constant and satisfying Equation (3.1)

Optimization leads to finding a stationary value of $\bar{J}$ (i.e., $d\bar{J} = \bar{0}$) for a given initial state x(0) and dx(0) = 0. Therefore, the stationary value of $\bar{J}$ is found if

$$d\bar{J} = \partial H^*/\partial u(0) = 0 \qquad (3.8)$$

Once the conditions satisfying Equation (3.8) are met, $\bar{J}$ can then be obtained. Equation (3.8) is called the Optimality Condition. Applying this condition to Equation (3.4) this condition states that

$$\frac{\partial H^*}{\partial u(0)} = \frac{\partial L^*}{\partial u}[\, x(0), u(0)\,] + \frac{\partial}{\partial u}\left[\lambda^T F^*[x(0),u(0)]\right] = 0 \qquad (3.9)$$

Multiple-Stage System. Progression from a single-stage system to a multiple stage system can be easily done by re-expressing Equation (3.1) as

$$x(i+1) = F^i[\, x(i), u(i)\,] \qquad (3.10)$$

where the initial state x(0) is given and i = 0, 1, ... N-1 with N ≡ nth stage. Here, Equation (3.10) describes a sequential set of equality constraints with x(i) as a

sequence of n-vectors to be determined by another sequence u(i) of m-vectors Schematically, this is expressed as a cascade system.

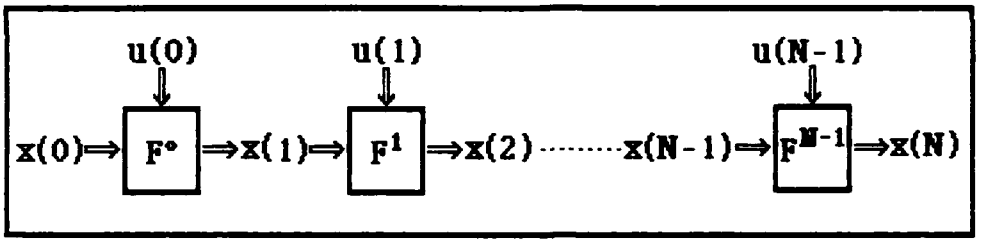


Fig. 3.3: Flow Chart for Multiple-Stage System

[Ref: 24]

The corresponding adjoined performance index J is

$$\bar{J} = \phi [x(n)] + \Sigma L^i [x(i), u(i)] + \lambda^T(i+1)\left\{F^i[\lambda(i),u(i) - \lambda(i+1)\right\}]. \qquad (3.11)$$

Using similar operation as for the single-stage case, the **optimality condition** is expressed as

$$\partial H^i / \partial u(i) = 0 \qquad (3.12)$$

where $$H^i = L^i [x(i),u(i)] + \lambda^T (i +1) \left\{F^i [x(i),u(i)] - l(i+1)\right\} \qquad (3.13)$$

for $i = 0,1, \ldots N-1$

Hence, to find a control vector sequence u(i) that produces a stationary value of the performance index $\bar{J}$. effort must be placed in solving the following **difference** equations:

$$x(i + 1) = F^i [x(i),u(i)] \qquad (3.14a)$$

and

$$\lambda(i) = [\partial F^i/\partial x(i)]^T \lambda(i +1) + [\partial L^i/\partial x(i)]^T \qquad (3.14b)$$

u(1) is that control sequence that makes H^i stationary. (From Equation 3.12) That is

$$\partial H^i / \partial u(i) = \partial L^i / \partial u(i) + \lambda^T(i+1)\ \partial F^i / \partial u(i) = 0 \qquad (3.15)$$

[REF: 24]

Application to Problem. The single-stage optimization can easily be applied to the present objective of maximizing the final semi-major axis. The problem parameters were established as follows:

let $u(0) = (\theta,\eta,\zeta,\phi-\psi) = (\theta,\eta,\zeta,PA)$ (3.16a)

$x(0)$ = initial state (3.16b)

$x(1)$ = final state (3.16c)

$x = [\, a\,,\, e\,,\, i\,,\Omega]$

$F^*[\ ] = [x_o + \Delta x]$ (3.16d)

$\lambda = [\, \lambda_a,\lambda_e,\lambda_i,\lambda_\Omega\,]$ (3.16e)

$L^* = 0$ (3.16f)

Applying equation (3.16) to the single-stage system optimality condition, the condition to be met is found to be

$$\partial H^*/\partial u(0) = \lambda^T(1)\ \partial F_a/\partial u(1) = 0 \qquad (3.17)$$

For maximizing the semi-major axis, the Langrange multiplier vector becomes

$$\lambda^{(1)} = [1, 0, 0, 0] \tag{3.18}$$

Since F^* constitutes the changes in the state vector, we can express it formally as.....

$$F^* = \begin{bmatrix} a_o + \Delta a \\ e_o + \Delta e \\ i_o + \Delta i \\ \Omega_o + \Delta\Omega \end{bmatrix} = \begin{bmatrix} F_a \\ F_e \\ F_i \\ F_\Omega \end{bmatrix} \tag{3.19}$$

Similarly, the control vector u(0) is expressed as

$$u(1) \quad [\theta^1, \eta^1, \zeta^1, (\theta+\psi)^1]$$

$$u(1) = [u_1^1, u_2^2, u_3^1, u_4^1]$$

$$u(1) = (u_i^1) \quad \text{for } i = 1,2,3,4 \tag{3.20}$$

By definition, the expression $\partial F^*/\partial u^1$ becomes (with PA = $\phi+\psi$),

$$\partial F^*/\partial u^1 = \begin{bmatrix} \partial F_a/\partial\theta & \partial F_a/\partial\eta & \partial F_a/\partial\zeta & \partial F_a/\partial PA \\ \partial F_e/\partial\theta & \partial F_e/\partial\eta & \partial F_e/\partial\zeta & \partial F_e/\partial PA \\ & & & \\ \partial F_i/\partial\theta & \partial F_i/\partial\eta & \partial F_i/\partial\zeta & \partial F_i/\partial PA \\ \partial F_\Omega/\partial\theta & \partial F_\Omega/\partial\eta & \partial F_\Omega/\partial\zeta & \partial F_\Omega/\partial PA \end{bmatrix} = [C] \tag{3.21A}$$

Since $\lambda^T(1)$ is not identically zero in equation (3.17), the following must hold true.

$$\partial Fa/\partial u^1 = [C] = [0]. \qquad (3.22A)$$

The scope of the problem is trmendously reduced by setting the performance index to be the optimum change is Δa per orbit. Interest then lies only in the function Fa which constitues the initial semi-major axis and its change. Furthermore, equation (3.21 reduces the complication one more degree by taking the partial derivatives. Since a_0 is the initial state and is constant, its presence in the differentiation can be ignored. The remainder is plain Δa. The scope of the problem in hence greatly reduced by an order of magnitude.

This means that equation (3.21A) can be reduced in size to the following.

$$\partial F_a/\partial u(1) = [\partial F_a/\partial \theta, \partial F_a/\partial \eta, \partial F_a/\partial \zeta, \partial F_a/\partial PA] \qquad (3.21B)$$

using the notation expressed in equation (3.20) this can be rewritten as

$$\partial F_a/\partial u(1) = [\partial F_a/\partial u_1, \partial F_a/\partial u_2, \partial F_a/\partial u_3, \partial F_a/\partial u_3] \qquad (3.21C)$$

or

$$\partial F_a/\partial u(1) = [\partial F_a/\partial u_1] \quad \text{for } i = 1,2,3,4 \qquad (3.21D)$$

The control vector u(1) that satisfied this Optimality Condition Equation (3.22) will determine the stationary value of the performance index J, where J is stated above as the maximum final orbit: J = a(N). For this single-stage, J = a(1) = the semi-major axis after 1 orbit. The search for this control vector is the quest of this effort. The solution approach follows.

3.2 Solution Approach.

Solution Formulation. F^* is given as a function of the control parameter (u_i). Specifically, looking at Fa only.

$$F_a = F_a(u_1, u_2, u_3, u_4) = F_a(u_i) \quad \text{for} \quad i = 1,2,3,4 \tag{3.23}$$

$$\text{where } (u_1, u_2, u_3, u_4) = (\theta, \eta, \zeta, \theta-\psi) \tag{3.24}$$

F_a is essentially a single function of four variables. The interest is in finding values of (u_i) at which F_a is a maximum. Under certain conditions, at such "places" in the space of variables (u_i),

$$[C] = [\partial F_a/\partial u_i] = [0] \quad \text{for} \quad i = 1,2,3,4 \tag{3.25}$$

One method of searching for this special (u_i) is as follows:

a. assume an initial guess u^* and that $[\partial F_a/\partial u_i^*]$ exist.

b. assume that $F_a(u_i) = 0$ can be expanded as

$$F_a(u_1,u_2,u_3,u_4) = F_a(u_1^*,u_2^*,u_3^*,u_4^*)$$

$$+ \sum_{j=1}^{n} \partial F_a/\partial u_j \,(u_1^*,u_2^*,u_3^*,u_4^*)\,(u_j - u_j^*)$$

$$+ \tfrac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \partial F_a/(\partial u_j \partial u_m)\,(u_1^*,u_2^*,u_3^*,u_4^*)\,(u_j - u_j^*)\,(u_m - u_m^*)\ldots \tag{3.26}$$

and $$\partial F_a/\partial u_i\,(u_1,u_2,u_3,u_4) = \partial F_a/\partial u_i\,(u_1^*,u_2^*,u_3^*,u_4^*)$$

$$\sum_{j=1}^{n} \partial^2 F_a/(\partial u_i \partial u_j)\; \partial^2 F_a/(\partial u_j \partial u_m)\,(u_1^*,u_2^*,u_3^*,u_4^*)\,(u_j - u_j^*) \tag{3.27}$$

Letting

$$[PC_{ij}] = \partial^2 F_a/(\partial u_i \partial u_j)(u_k^*) = \partial C/(\partial u_i u_j)(u_k^*) \tag{3.28}$$

in Equation (3.27) and using Equation (3.25), gives

$$u_i = u_i^* - \sum_{j=1}^{n} [PC_{ij}]^{-1}[C](u_1^*, u_2^*, u_3^*, u_4^*). \tag{3.29}$$

Defining $\delta u = (u_1 - u_1^*)$, (3.30)

Equation (3.29) can be expressed as

$$\delta u = -\sum_{j=1}^{n} [PC]_{ij}^{-1}[C](u_1^*). \tag{3.31}$$

Understanding that the summation must be taken, a simplified expression can be stated as

$$\delta u = -[\partial C/\partial u]^{-1}[C](u_i^*). \tag{3.32}$$

In Equation (3.32), $[C](u_i^*)$ denotes the Optimality Vector evaluated at the initial guess control vector u_i^* for $i = 1,2,3,4$.

δu denotes the required change in control vector u that would satisfy the condition for a stationary performance index J. Iterations of an "initial guess" is necessary until this $\delta u \approx 0$. A reduction of the iteration frequency can be achieved by providing a "good" initial guess. A graphical method of determining this guess is presented in Section IV under Surface Representation.

Analytical VS Numerical. Solution to Equation (3.32) requires the evaluation of the partial derivatives of four state parameter functions (Fa, Fe, Fi, F_{Ω}) with respect to four control variables (ϕ, η ,ζ ,PA). This involves the determination of a [4x4] matrix and its derivative. Recall, the objective is to find the control parameters leading to a maximum change in the semi-major. From the C Equation (3.21), it is apparent that interest lies in the 1st row elements. This row constitutes the optimality vector. This reduces the main effort considerably. This simplification provides the following:

$$[C] = [\partial\Delta a/\partial\,\theta \;\; \partial\Delta a/\partial\eta \;\; \partial\Delta a/\partial\zeta \;\; \partial\Delta a/\partial PA] \qquad (3.33)$$

Disguised in Δa are horrendous amounts of differentiation that can easily lead to any amount of errors. The choice made here was to circumvent the series of differentiations and use numerical techniques to determine the elements in Equation (3.26). To appreciate this approach, one must initiate the series of differentiation. Appendix B provides the reader this insight. The reason for the numerical method choice and why the analytical approach is avoided is apparent.

3.3 Numerical Formulation and Model.

The Numerical Models. As a first approximation to the partial derivative of the multi-variable function F(ui), the numerical differentiation techniques called Newton's Forward and Backward were used. These are expressed as...

$$F'(u_i) = \frac{F(u_i + \delta u_i) - F(u_i)}{\delta u_i} \qquad \text{(Forward)} \qquad (3.34)$$

$$\text{and} \quad F'(u_i) \approx \frac{F(u_i) - F(u_i + \delta u_i)}{\delta u_i} \qquad \text{(Backward)} \qquad (3.35)$$

For a simple two-variable case, these can be interpreted as the slopes of the two lines shown in Figure 3.8.

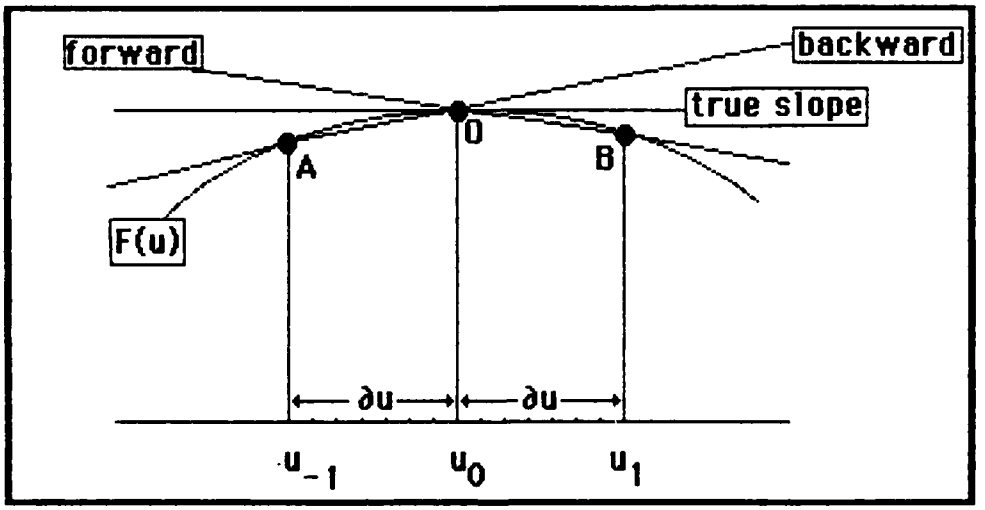


Fig. 3.8: Derivative Approximation Models

Initial Results. Initial Results were not encouraging. Initial intuitive expectation were filled with optimism for a semi-well behaved derivative of the function $F(u_i)$. This was not the case. These formalas displayed a "wandering" type behavior with the approximated derivatives for each iteration. The Forward Formula can be re-expresssed as

$$\Delta u \approx \{F(u)_{new} - F(u)_{old}\} / \delta u \approx - F(u)/F'(u) \qquad (3.46)$$

Using this formulation, the expectation was for some kind of rough convergence. Instead, the Δu_i were fluctuating after each updated estimate $F(u_i)$new.

Sources of Errors. A study of test cases suggested that approximate derivatives obtained from from such polynomial $F(u_i)$ be viewed with skepticism unless very accurate data are available. Even with accurate data (or initial guess as it is commonly called), the accuracy diminishes with increasing order of the derivatives. This is the problem dealt with the second order derivatives . The dominant error source is in the input errors. These proved very critical. Even when the initial guess was close to the theoretical value, the input was still very critical because the approximating (along with the perturbation) algorithms magnify them enormously. The crucial factor seemed to be the magnitude of δu. The magnification of input error behaved inversely to this value whereas the inevitable truncation error was directly affected.

Improved Model. The models intially used were abandoned after countless futile attempts to contain the "wandering" derivatives. From Figure 3.8 it was evident that a more accurate approximation would be achieved by using the slope of a line connecting points A and B. This approach is commonly called the Stirling Method.

It is basically the Newton method modified. This well-known formula also known as central differencing is expressed as

$$F'(u) \approx \{F(u + \delta u) - F(u - \delta u)\} / (2\delta u) \qquad \text{(Stirling)} \qquad (3.47)$$

Numerical Formulation. This formula was employed in determining the required derivative expressed in Equations (3.44) of the previous section. Appendix B shows how it was coupled to accomodate a function of four variables.

Results. The use of Stirlings Method provided better results. The "wandering" phenomenon experienced with the previous Newton's methods was greatly diminished by an order of magnitude.

From the perspectives peak points were selected as test cases. Known maxima is inputed. With the appropriate coordinates for the maxima (from perspectives as input, convergence to within. .1 degree was achieved. This related the fact that a maximum did indeed exist within the area of search. This gave additional credence to the perspectives. The various identifiable coordinates were checked with similar results. Convergence was achieved in two iterations. The bottom line was that the required calculated change in control parameters was within .1 degree a change so small that virtually a maximum was discovered. This was satisfaction but not with surprise since approximate coordinates of the maxima were inputted. The real test follows.

Approximating $\partial 2F/(\partial ui \partial ui)$. The Stirling Method is used to approximate the $\partial^2 F/(\partial u_i \partial u_j)$ matrix. For simplicity, the following definition is made:

$$\text{Let} \quad [PC_{i,j}] = \partial^2 F/\partial u_j \partial u_i = \partial C/\partial u_j \tag{3.36A}$$

$$\text{where} \quad C = \partial F/\partial u_i \tag{3.36B}$$

The subscripts i and j indicate a particular control vector from the set $(u_1, u_2, u_3, u_4) = (\theta, \eta, \zeta, PA)$. Recall, that for an initial guess control vector, F is defined as

$$F = F(u_1^*, u_2^*, u_3^*, u_4^*) \tag{3.37}$$

where the superscript (*) denotes the initial guess. Note that for if any iteration is desired, the u_i^* term would be the updated value which would become the new initial guess.

Hence, $(u_1^*, u_2^*, u_3^*, u_4^*) = (\theta^*, \eta^*, \zeta^*, PA^*) \equiv$ initial guess.

With this notation, equation (3.36) can easily be expanded using Stirlings Method.

Example. To show how the matrix is constructed, the $PC_{1,2}$ component is selected as an example.

$$PC_{1,2} = \frac{\partial^2 F_a}{\partial u_1 \partial u_2} \tag{3.38}$$

$$PC_{1,2} = \frac{\frac{\partial F_a}{\partial u_2}(u_1^* + \delta u_1, u_2^*, u_3^*, u_4^*) - \frac{\partial F_a}{\partial u_2}(u_1^* - \delta u_1, u_2^*, u_3^*, u_4^*)}{2\delta u_1} \tag{3.39}$$

Expanding the $\partial F_a / \partial u_2$ terms via the same method, this equation becomes

$$PC_{1,2} = 1/2\delta u_1 \left\{ \frac{F_a(u_1^* + \delta u_1, u_2^* + \delta u_2, u_3^*, u_4^*) - F_a(u_1^* + \delta u_1, u_2^* - \delta u_2, u_3^*, u_4^*)}{2\delta u_2} - \frac{F(u_1^* - \delta u_1, u_2^* + \delta u_2, u_3^*, u_4^*) - F(u_1^* - \delta u_1, u_2^* - \delta u_2, u_3^*, u_4^*)}{2\delta u_2} \right\} \tag{3.40}$$

Gathering terms and simplifying, Equation (3.40) can be expressed as

$$PC_{1,2} = \frac{\left| F(u_1^* + \delta u_1, u_2^* + \delta u_2, u_3^*, u_4^*) - F(u_1^* + \delta u_1, u_2^* - \delta u_2, u_3^*, u_4^*) - F(u_1^* - \delta u_1, u_2^* + \delta u_2, u_3^*, u_4^*) - F(u_1^* - \delta u_1, u_2^* - \delta u_2, u_3^*, u_4^*) \right|}{4\delta u_1 \delta u_2} \tag{3.41}$$

For the diagonal elements (i = j), the case (i = 1, j = 1) is shown below:

$$PC_{1,1} = \partial^2 F_a / (\partial u_1 \partial u_1) = \partial^2 F_a / (\partial u_1)^2 \tag{3.42}$$

$$PC_{1,1} = \frac{F_a(u_1^* + \delta u_1, u_2^*, u_3^*, u_4^*) - 2\,\partial F_a(u_1^*, u_2^*, u_3^*, u_4^*) + F_a(u_1^* - \delta u^*, u_3^*, u_4^*)}{4\,\delta u_1^{\,2}} \tag{3.43}$$

By following the same procedure for the other elements, [PC_{ij}] matrix can be constructed as follows:

$$[PC_{i,j}] = \begin{bmatrix} \partial^2 F_a/(\partial u_1)^2 & \partial^2 F_a/(\partial u_1 \partial u_2) & \partial^2 F_a/(\partial u_1 \partial u_3) & \partial^2 F_a/(\partial u_1 \partial u_4) \\ \partial^2 F_a/(\partial u_2 \partial u_1) & \partial^2 F_a/(\partial u_2)^2 & \partial^2 F_a/(\partial u_2 \partial u_3) & \partial^2 F_a/(\partial u_2 \partial u_4) \\ \partial^2 F_a/(\partial u_3 \partial u_1) & \partial^2 F_a/(\partial u_3 \partial u_2) & \partial^2 F_a/(\partial u_3)^2 & \partial^2 F_a/(\partial u_3 \partial u_4) \\ \partial^2 F_a/(\partial u_4 \partial u_1) & \partial^2 F_a/(\partial u_4 \partial u_2) & \partial^2 F_a/(\partial u_4 \partial u_3) & \partial^2 F_a/(\partial u_4)^2 \end{bmatrix}$$

(3.44)

This matrix is used in computing the change in control parameters necessary to determine the stationary value of the performance index J. From Equation (3.),

$$\Delta u_i = u_{new} - u_{old} = -(C^*)/[C'] \quad \text{for } i = 1,2,3,4 \tag{3.45A}$$

where (C^*) denotes the $\partial F_a/\partial u_i$ evaluated at the initial control values and [C'] denotes the $\partial^2 F_a/(\partial u_i \partial u_j) = [PC_{ij}]$. The C notation is later used in the algorithm developed to compute the Δu_i. Hence,

$$\Delta u_i = u_{new} - u_{old} = -(C^*)/[PC_{ij}] \tag{3.45B}$$

Note that (C^*) is a four-element row vector and [PCij] is a [4x4] matrix. Performing the appropriate operation, the required change in the control vector sought here is plainly,

$$\Delta u_i = u_{new} - u_{old} = -[PC_{ij}]^{-1}(C^*) \tag{3.45C}$$

A stationary value is achievable if there exist a control variable set u_i such that the Δu_i is approximately zero. That is,

$$\Delta u_i = u_{new} - u_{old} = - [PC_{ij}]^{-1} (C^*) = (0) \qquad (3.45D)$$

This would indicate that there are no changes necessary in the previous control variables set to arrive at the optimum Δa. Such is the particular I set of controls that would render the performance index J stationary.

Software Development. Equation (3.45D) is the basis of the software that is developed to find the stationary value of the Performance Index J. The process involves the coding of the perturbation solution equations presented by Jenkins [Ref:10]. The Δa equation is the only equation generating the required vectors and matrices in Equation (3.4) since the objective solely called for maximizing this function. The incorporation of the other state functions, as expressed in Equation (3.19), can readily be employed with little transitional difficulty. The mechanism of the search process included several iterations to construct the elements of (C) vector and [PC] matrix. Once this [PC] matrix was found, a specially formatted program is used to compute its inverse PCI. Equation (3.45D) is then incorporated to determine the neccesary changes in the control vector u. The process continues until a specified convergence criterion is established. The interactive program outputs all the computed (intermediate and final) vectors and matrices along with the required Δu_i values.

3.4 The Nature Of The Δa Function.

The change in the semi-major axis is the main objective of this study. It almost seems as a trivial task until the function itself is confronted. Δa is a function of four variables. $\Delta a = \Delta a(\theta,\eta,\zeta, \phi\text{-}\psi)$. The main difficulty in maximizing this function analytically is with the complexity of the derivative of the function (as was explained in section 3.1 and Appendix B). The wise choice of employing numerical techniques to determine the maxima of Δa still presents another difficulty-the initial guess. This initial quess becomes the starting point in the search of the set of control vector which maximizes Δa. Due to the search pattern of the modified Newton-Rhapson (or Stirling Method), this initial guess becomes critical and must serve as a "good" starting point. This is where the difficulty begins: What is a "good initial guess"? How can a region of "good guesses be established?" The answers can be provided by a surface representation of the function Δa. The next section explains an In-House capability available through UNIX/VAX system that provides this surface representation.

3.5 Surface Representation.

A canned plotting package called "S-Package" residing in the UNIX/VAX system is used to provide three-dimensional perspectives of the Δa function. This program provides an quasi-isometric three-dimensional perspective of an output function with respect to two input variables with the output function as the third dimension. Analyzing Δa is accomplished by holding the angular momentum orientation angles (η and ζ) stationary while allowing variations in the coning angle (θ) and the phase angle ($PA = \phi\text{-}\psi$). The resulting perspectives show the behavior of the Δa function as these selected parameters are varied. These are presented in more detail in the Results section. Such surface representations describing the behavior of the change in semi-major axis with respect to a coning Solar Sail have never been seen before!

These perspectives, provide very valuable information on the nature of the Δa function as certain orientations are approached. More importantly, the perspectives serve as a source of information on the region of search from which one can determine a "good initial guess". The maxima and minima are apparent. Additionally, problem areas that might cause a gradient search scheme to wander excessively and perhaps fail can be avoided simply by identifying critical areas from the perspectives.

No matter what scheme is used to present a function of more than two, there still remains the question of the behavior of all the variables. Man, limited to working with three dimensions, representing a function of more than three variables is quite a task. In this study, Δa is a function of four variables; therefore, a five-dimensional perspective is ideal to explicitly show its true behavior with respect to all the variables. Accepting this limitation and expressing the function with respect to three-dimensions is the best anyone can presently do. Despite the limitation, the benefits of having at least a five-dimensional representation is overwhelming.

IV. Results

4.1 Surface Perspectives.

The nature of the Δa function is analyzed for the occurrence of maxima as the various control parameters (θ, η, ζ, PA) are varied. Being a multi-variable function, it becomes practically impossible to look at the overall nature of this Δa function as all the parameters are allowed to vary. It is instructive to hold any two less dynamic variables and perspectively look at the function with the two more dynamic variables allowed to go free. This would constitute a three-dimensional representation of the Δa function. Abundant information can be read off such perspectives since all the variations and and resultant behavior of the function can be readily seen in a given perspective. Anomalies are conspicous by their presence (if any are present). The extrema (maxima and minima) can easily be spotted and located for the conditions given. It is also very helpful in identifying areas of "stagnation" or "stability" in which the function can very well remain stationary even with small perturbations. This very fact that one has a three-dimensional interactive look at the behavior of the function makes a valuable preliminary tool in the optimization process where attaining a "good initial guess" is so paramount. Such information is available with perspectives. Such dynamic information is found absent in tabular formatted data.

The information contained in the perspectives found in this study contributes to the search scheme developed to find the optimal control setting angles to achieve an optimum change in the semi-major axis per orbit. The "initial guess" barrier has just been broken.

Variation in Inclination. Interest exists in the behavior of the change in the semi-major axis Δa due to changes in the sailcraft's orbit inclination. Figure 4.1 shows the relative Δa magnitude of three select inclination with changing coning angles.

angles.

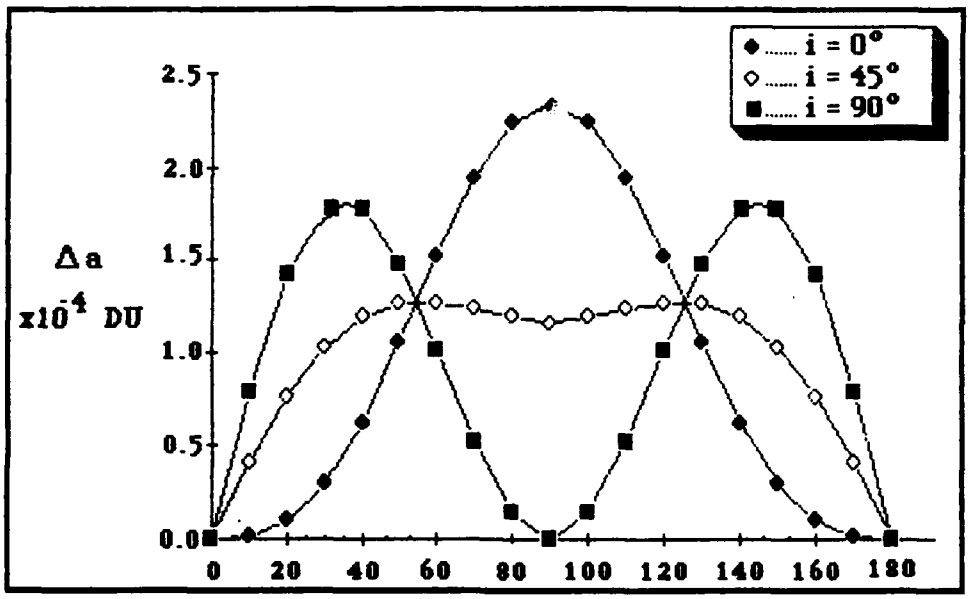


Fig. 4.1: Coning Angle (0°)

Note the decreasing Δa magnitude as the inclination increases and then its subsequent increase to form two maxima as the inclination is greater than 45°. This behavior was graphically evaluated more closely as inclination varied from 0 to 90 degrees. Figures 4.2A through 4.2G (presented in the following pages) show the peculiar behavior of this Δa function as the inclination is increased. Each figure depicts a different case with with its corresponding Δa magnitude. Although these relative Δa magnitude differences are not shown in these figures, the maximum Δa experienced per case is indicated. The important information is in the overall behavior of this change in semi-major axis. The intersesting aspects of this Δa behavior are summed as follows:

As the inclination is increased from 0° to 90°,

a. the magnitude of Δa decreases as the inclination approaches 45 degrees.

b. the magnitude of Δa increases as the inclination becomes greater than 45 degrees but does not attain its original maximum value.

c. the local maximum shifts location away from $\theta = 90^\circ$ (at $i = 0^\circ$) to form two maxima at $\theta_1 = 35^\circ$ and $\theta_2 = 145^\circ$ (at $i = 90^\circ$);

d. maxima occur at $PA = -90^\circ$ and remains stationary at this value, i.e., it does not vary with changes in inclination;

The three select cases are further examined in the following pages to show why such particular control angles work to maximize Δa. The three cases looked at are as follows:

Case 1: 0° inclination

Case 2: 45° inclination

Case 3: 90° inclination

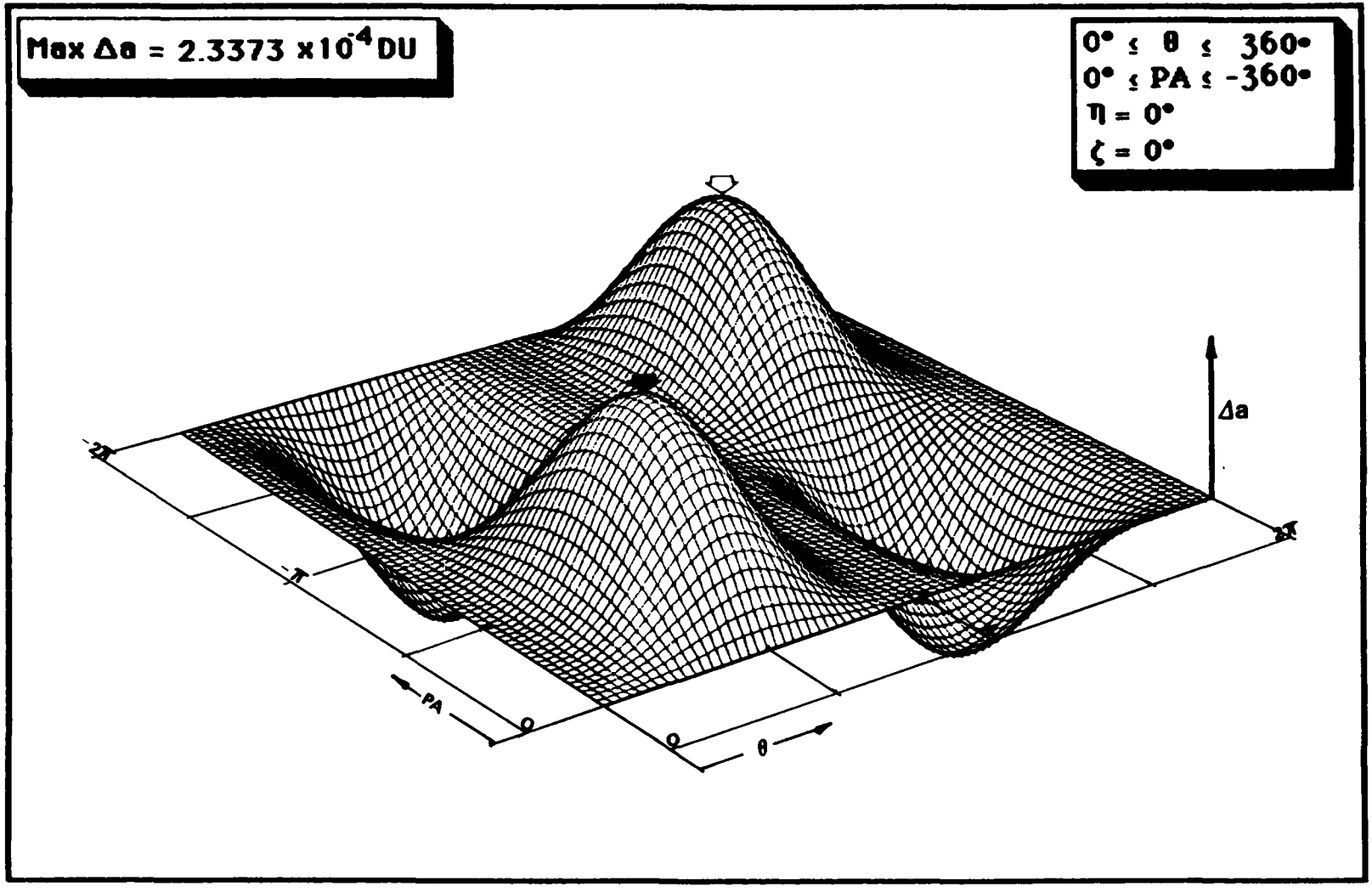


Fig. 4.2 A: Perspective (i = 0°)

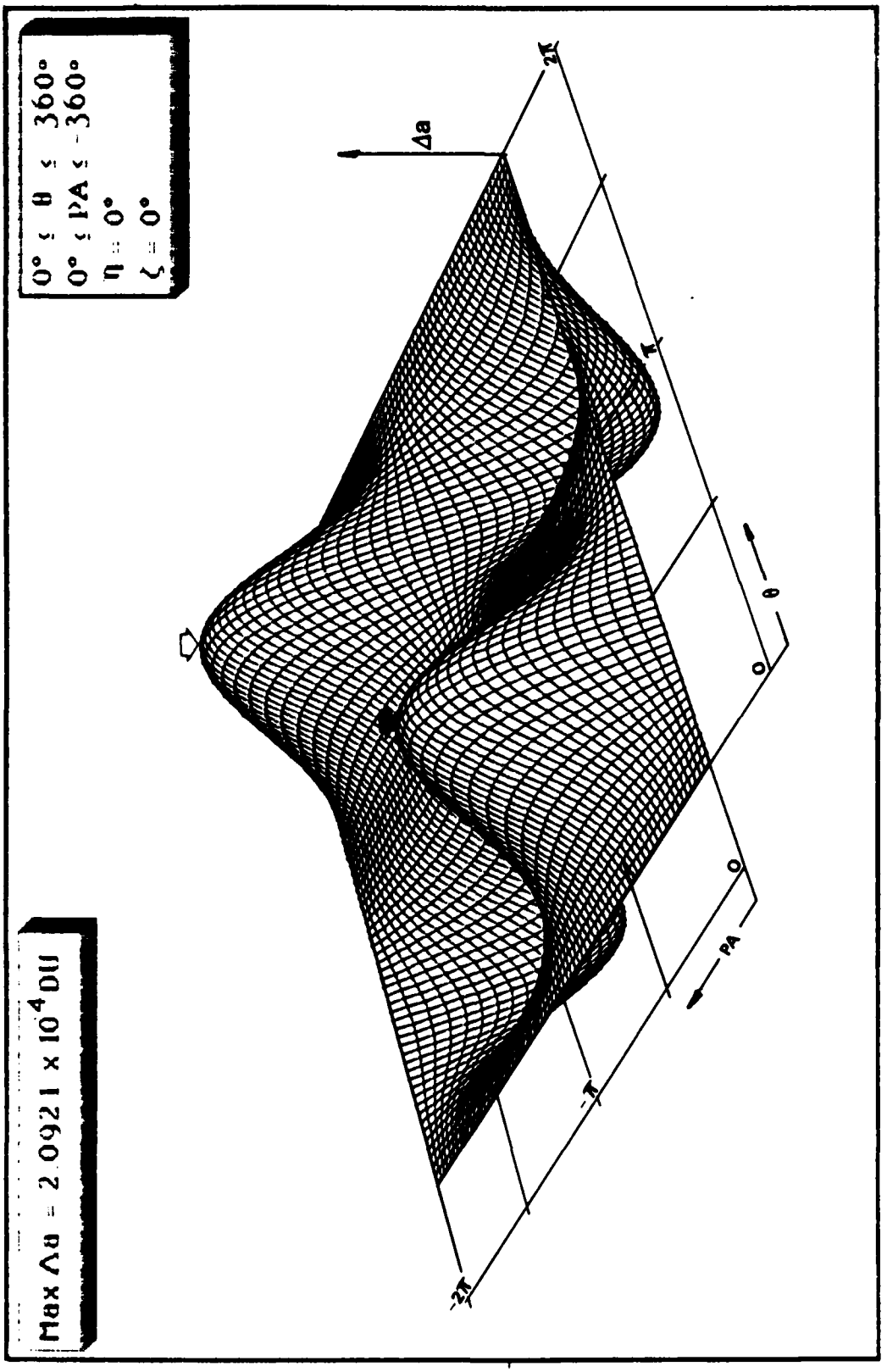


FIG. 4.28 Perspective (i = 15°)

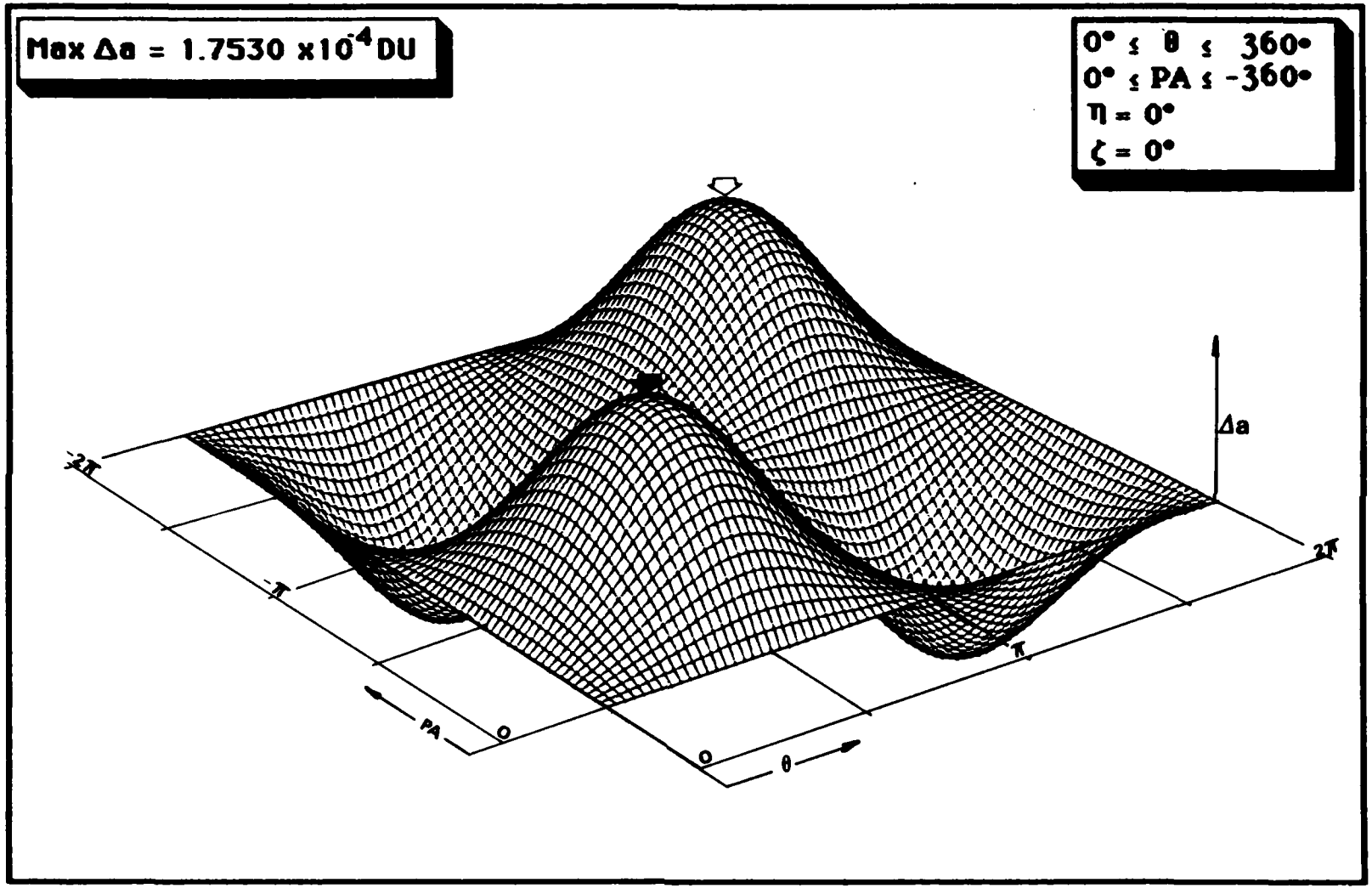


Fig. 4.2C : Perspective (i =30°)

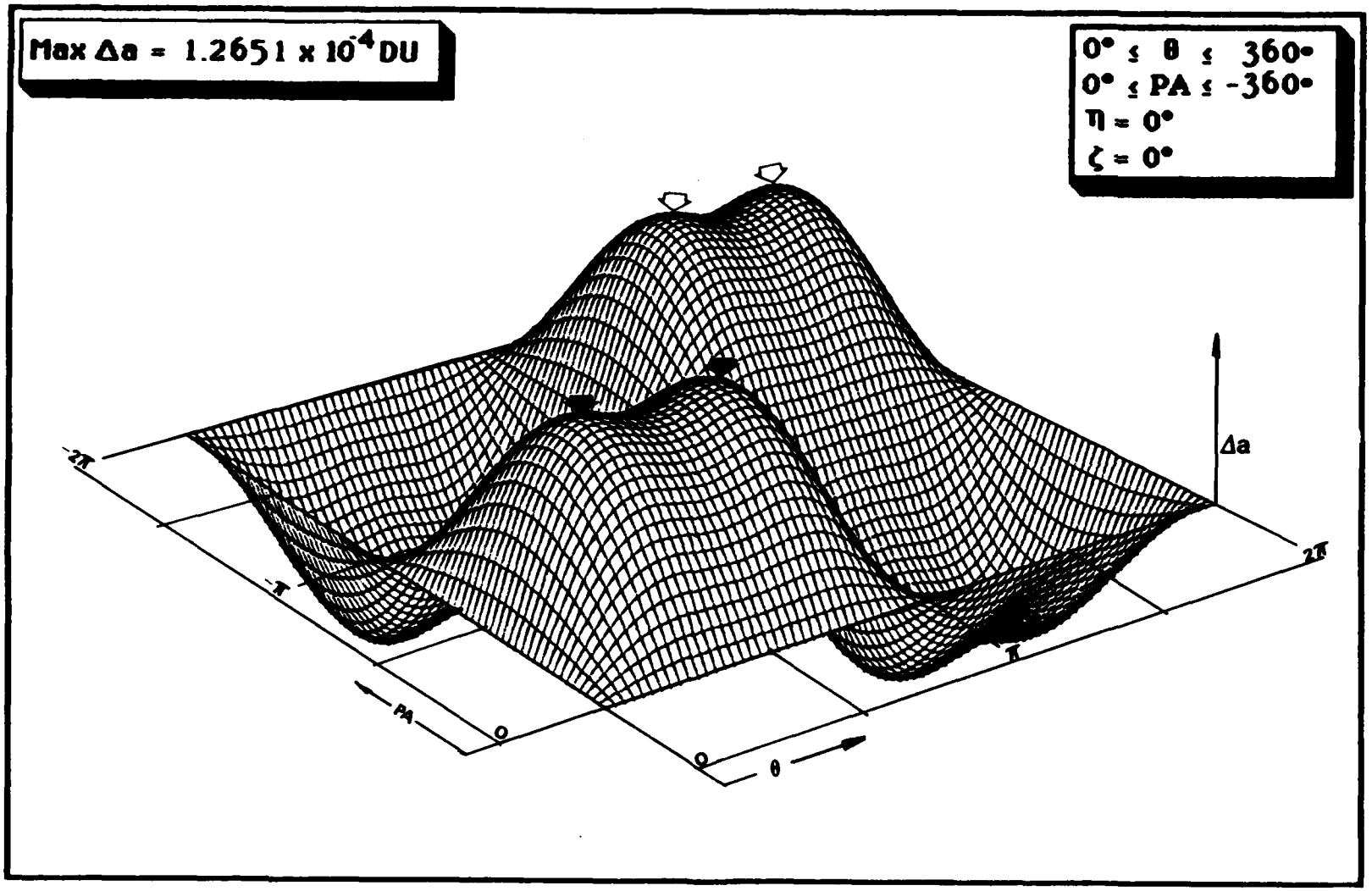


Fig. 4.2D: Perspective (i = 45°)

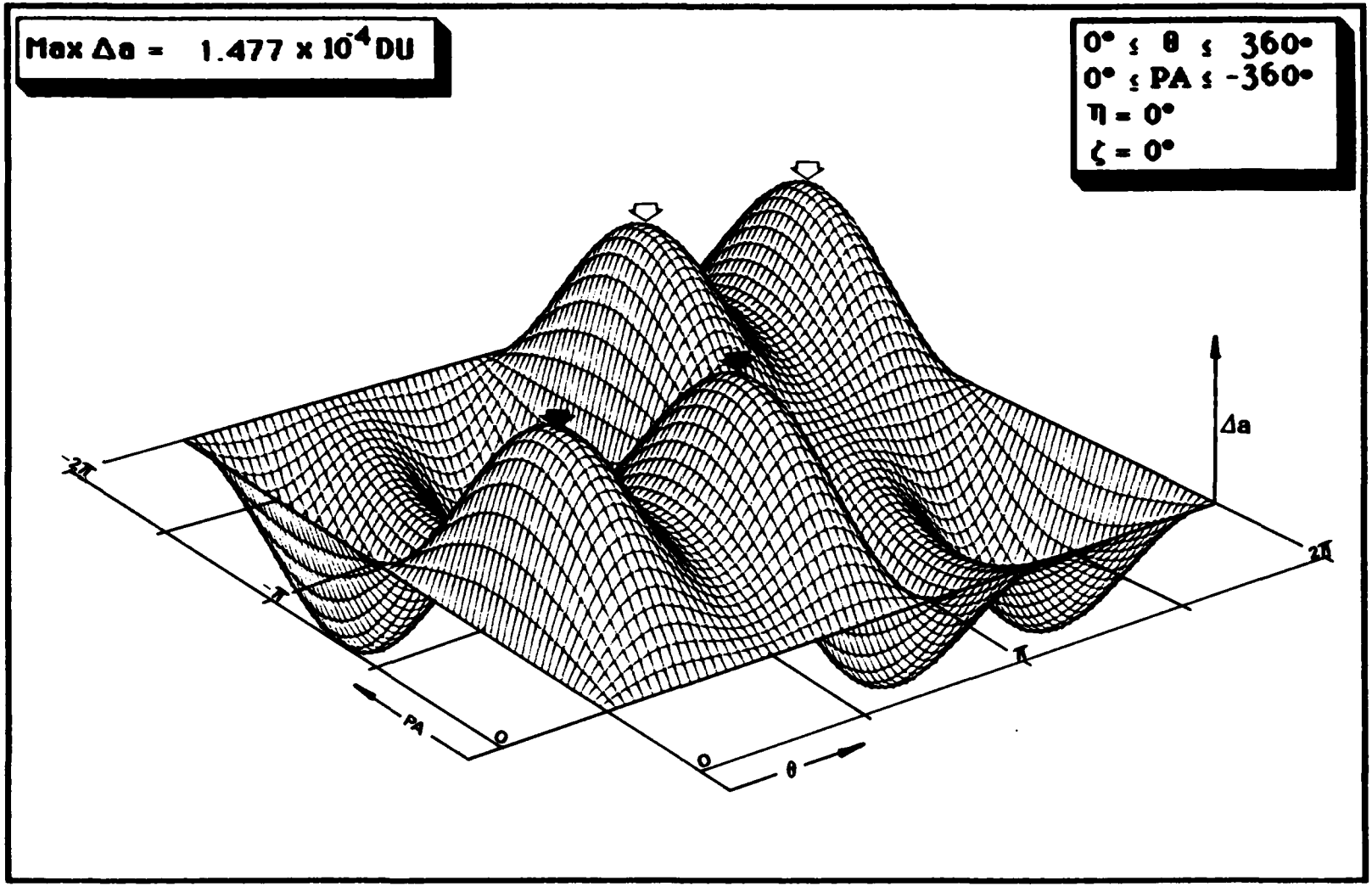


Fig. 4.2E : Perspective (i = 60°)

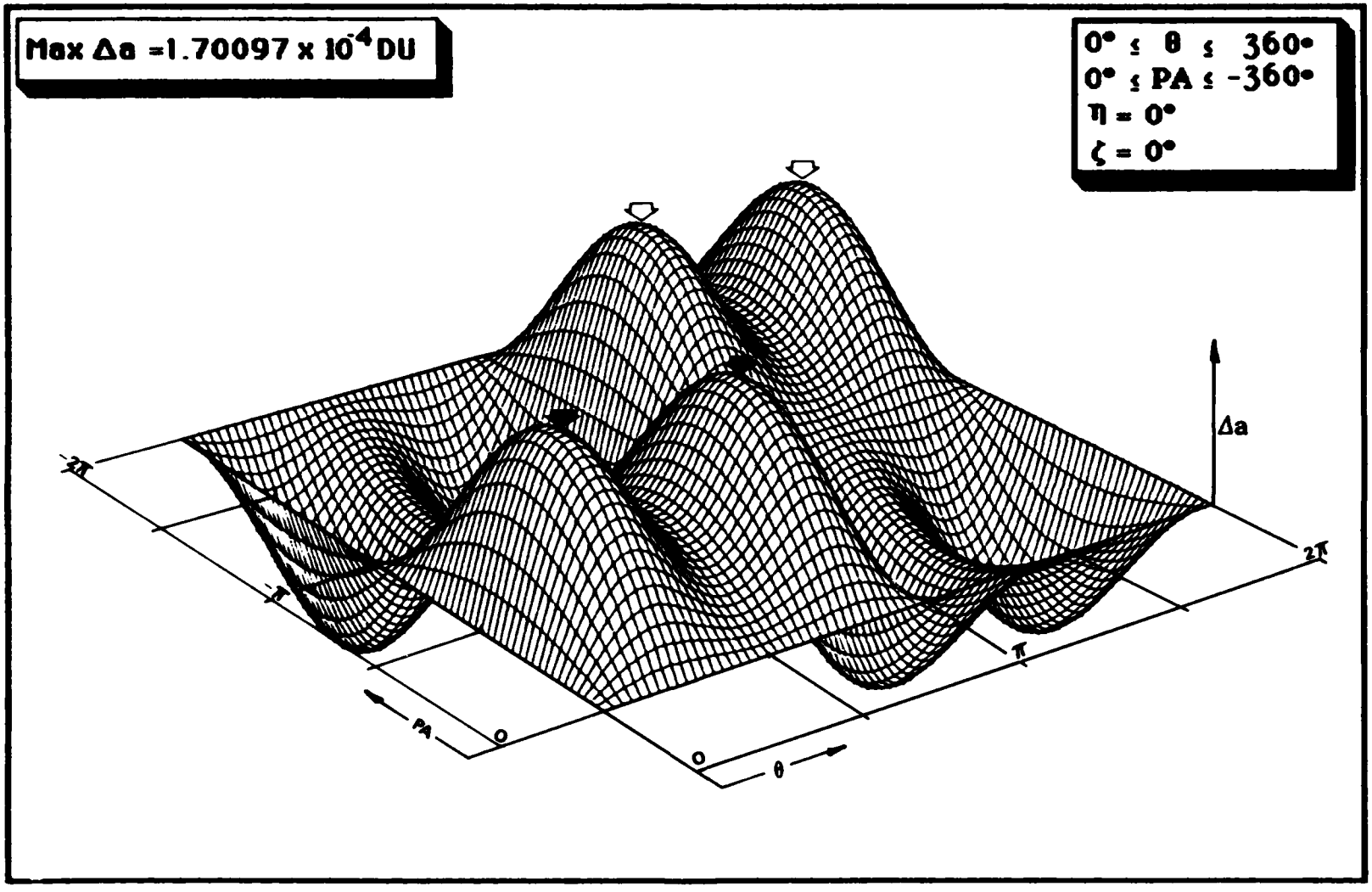


Fig. 4.2F : Perspective (i = 75°)

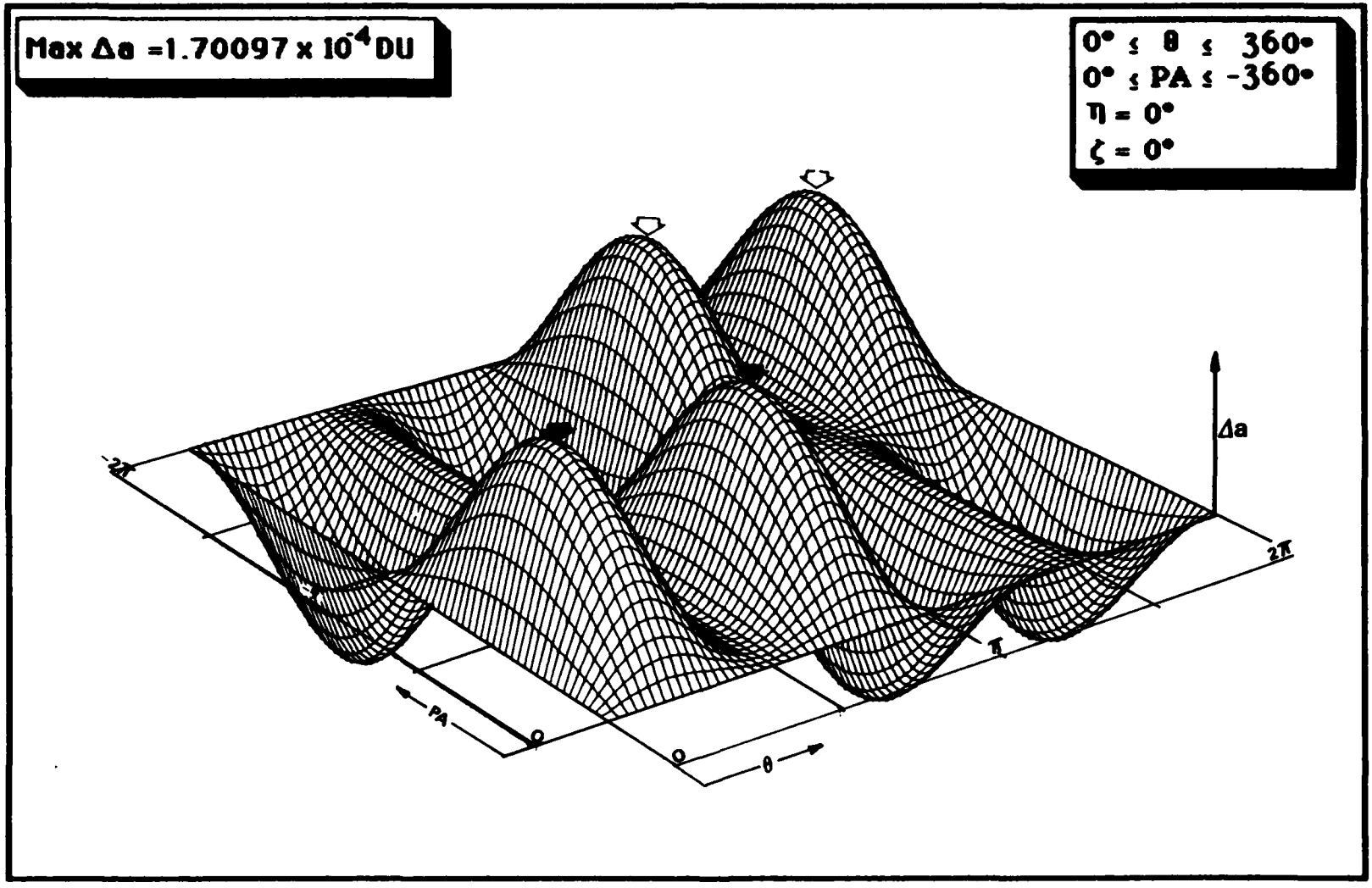


Fig. 4.2G: Perspective (i = 90°)

0 Degree Inclination. Motion in the ecliptic plane with the coning angle set at 90° and the phase angle at -90° has produced the largest change in the semi-major axis at 2.337×10^{-4} DU. Figures 4.1 compares the behavior in this inclination with other inclined orbits; Figure 4.2A perspectively portrays the behavior of the Δa function as any of the other control parameters (θ and PA) are varied. For a coning angle of 90°, the solar sail is (at certain intervals in the orbit) perpendicular to the solar radiation; in such an orientation, there is maximum thrust produced in the direction away from the sun. However, maximum thrust does not always equate to maximum Δa. It could very well work against maximizing Δa; such a case is when the thrust vector has a component working against the orbital velocity vector. Correct phasing angle PA allows the optimum use of maximum thrust conditions leading towards maximizing Δa.

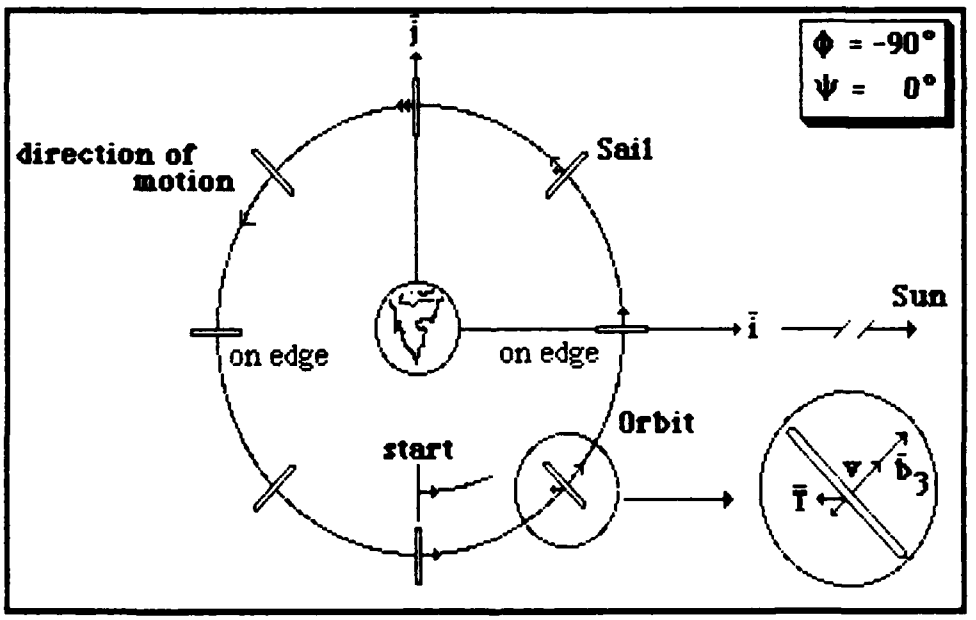


Fig. 4.3: 0° Inclination

This PA parameter relates the initial point of the sailcraft in the orbit and its initial orientation wrt the Body-centric Inertial IJK Frame. For θ - 90°, the latter information

(proper phase angle) is inconsequential since the sail is in a pure spin motion. However, for any other non-pure spin alttitude, this information is of critical importance. Hence, angle PA only relates where the sailcraft is in the orbit for this pure spin configuration. At PA= -90°, the sailcraft is configured as shown above in Figure 4.3; Note where the sailcraft is allowed to start its journey from: 90° counter-clockwise from the i axis. This orbit configuration produces the most Δa (Δa = 2.337 x10^{-4}Du) the torque-free solar sail can muster optimally at improving its radial distance from the planet earth.

45 Degree Inclination. As seen earlier, Figure 4.1 provided an indication of what would happen to Δa when the orbit inclination is increased. The shift of the single maximum at (θ, PA) = (90,-90) to form a pair of maxima can be very clearly seen in the perspective, Figure 4.2D. No longer does θ at 90 degrees monopolize the maximum point at this inclination: the two maxima occur at (55°, -90°) and (125°, -90°). This is important to understand: the pure spin configuration, though it may present the most exposed surface, does not present the optimum condition.

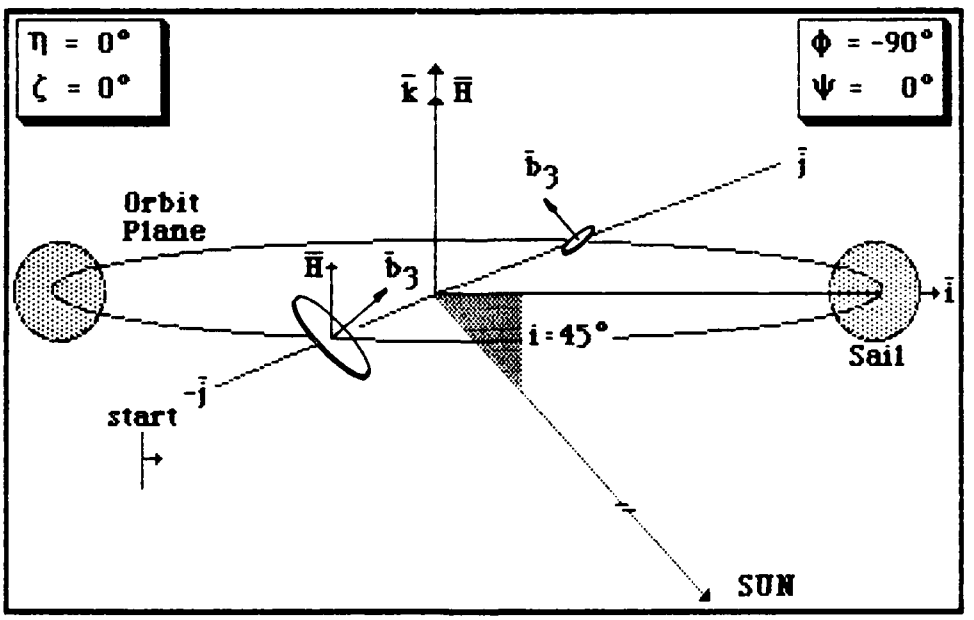


Fig. 4.4A: 45° Inclination

Figure 4.4A above shows the various positions in the orbit relation to the attitude of the solar sail wrt the sun. Only the 55° coning angle case is depicted here. The 125° coning case contributes the same effect. This is studied next.

The relationship between the two coning options are investigated to determine the resulting components of thrust along the velocity vector. Both diagrams shown here in Figure 4.4B relate to the position of the sailcraft at its starting position. Since the sailcraft's precession rate is equated to its orbital mean motion, the coning motion will be cyclic and in phase with the orbital period.

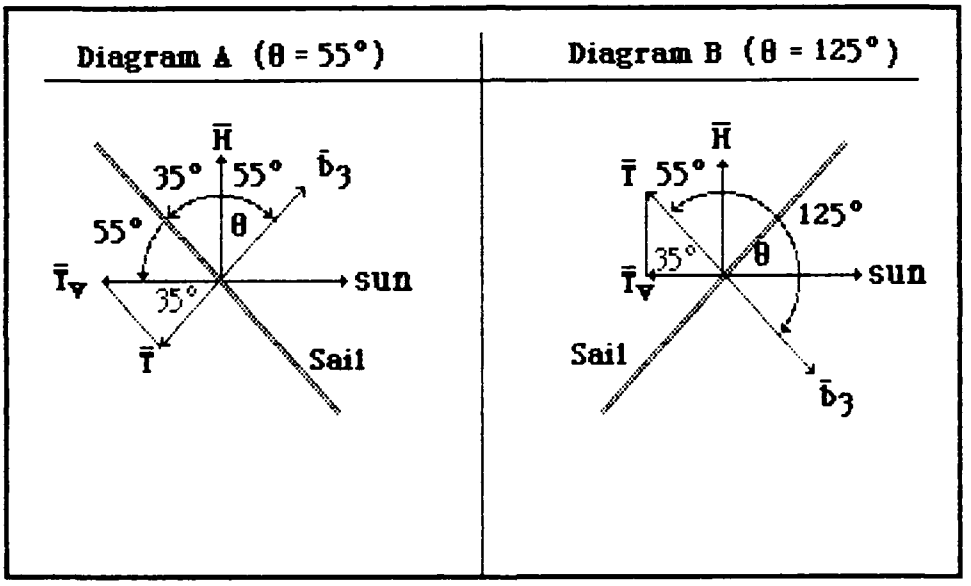


Fig. 4.4B: Coning Angle Comparison

$$\left.\begin{array}{ll} \text{From Diagram A:} & T_V = T\cos 35^\circ = .819\,T \\ \text{From Diagram B:} & T_V = T\sin 55^\circ = .819\,T \end{array}\right\} \Rightarrow T_V(\theta - 55^\circ) = T_V(\theta - 125^\circ)$$

Since the two components are identically equal, their corresponding Δa effects will also be equal. For the configurations of higher inclinations, this offers an option in the choice of coning angle.

90 Degree Inclination. This special inclination is depicted in Figure (4.5) below. Fimple [Ref: 5] states that this orientation affords the sailcraft continuous energy increase throughout the orbit. However, the magnitude of Δa is smaller than for that seen in the 0° inclination case because of the small thrust component in the direction of the orbital velocity vector.

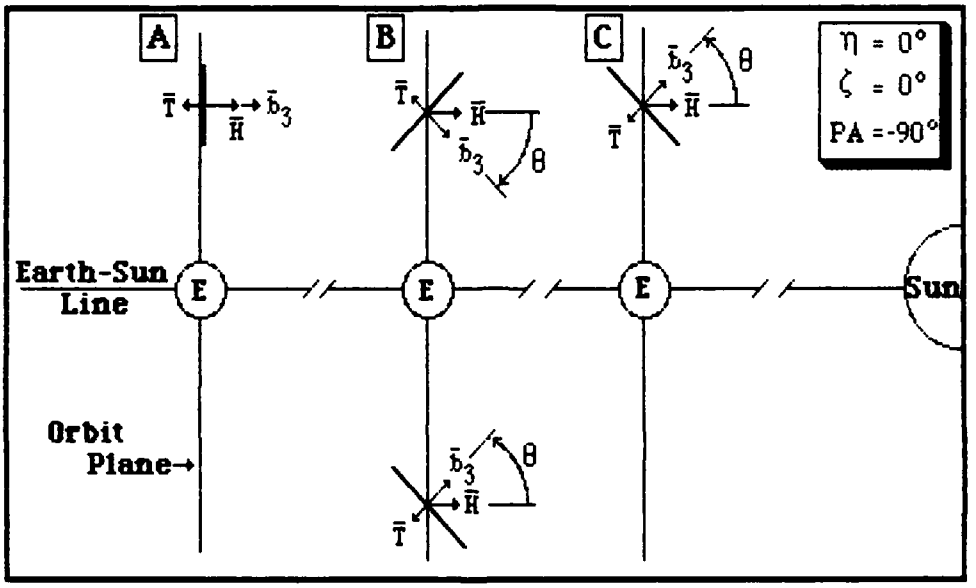


Fig.4.5: 90° Inclination

Figure 4.5 shows three control setting configurations with the angular mementum vector $\bar{H}$ parallel to the orbit plane normal. This then requires $\bar{H}$ to stay fixed towards the sun. Configuration A shows a coning angle of 0° and no thrust component in the orbital velocity vector direction; therefore, Δa equals zero. The corresponding perspective for this 90° inclination (Figure 2.4G) clearly shows this fact: As the coning angle is increased, a maximum Δa is achieved at $\theta = 35°$ and at $\theta = 145°$. The angle of 35° coincides with the results of a study done earlier by Garwin [Ref: 5] in which he finds that a tilt angle of ≈ 35 degrees provides the largest thrust component along the

velocity vector (see Section II. Previous Efforts). Given the similar orientation Garwin refers to, this tilt angle is just the coning angle ϑ of this study. Tsu [Ref: 19] also arrives at this optimum titl angle of 35 degrees. Configuration B depicts this control setting. For comparison with the same setting but at a different phase angle (PA = -270°), the thrust component is no longer working to maximize Δa, but working to minimize Δa. In fact, the minimum Δa is experienced with this control setting.

Variation in PA. If the variation in θ is restricted $0 < \theta < 360^{\circ}$, a maximum Δa occurs at PA equal -90°. Such occurence at this value can be seen analytically by studying Equations (2.21) and (2.22) shown below:

$$\Delta a = a^{1.5} D\, TP/\mu \left[da_1 + da_2 + da_3 + da_4 + da_5 + da_6 \right] \qquad (2.21)$$

$$da_1 = \tfrac{1}{2} D_1^{\,2} [(\beta 2+3\beta 4)\cos(\phi-\psi) - (3\beta 1+\beta 5)\sin(\phi-\psi)] \qquad (2.22a)$$

$$da_2 = \tfrac{1}{2} D_1 D_2 [(\beta 2-\beta 4)\sin(\phi-\psi) + (\beta 5-\beta 1)\cos(\phi-\psi)] \qquad (2.22b)$$

$$da_3 = 2 D_1 D_3 [\beta 6 \cos(\phi-\psi) + \beta 3 \sin(\phi-\psi)] \qquad (2.22c)$$

$$da_4 = 2 D_2 D_3 [\beta 3 \cos(\phi-\psi) + \beta 6 \sin(\phi-\psi)] \qquad (2.22d)$$

$$da_5 = \tfrac{1}{4} D_2 D_2 [(3\beta 2+\beta 4)\cos(\phi-\psi) - (\beta 1+3\beta 5)\sin(\phi-\psi)] \qquad (2.22e)$$

$$da_6 = D_3^{\,2} [(\beta 2+\beta 4)\cos(\phi-\psi) - (\beta 1+\beta 5)\sin(\phi-\psi)] \qquad (2.22f)$$

From Equation (2.22), one can say that maximizing the summation of the da terms equates to maximizing Δa. A plot (see Figure 4.6) of these da terms above for variations in the PA (recall, PA = $\phi-\psi$) component explicitly shows that the summation is indeed maximum at PA equal to -90 degrees.

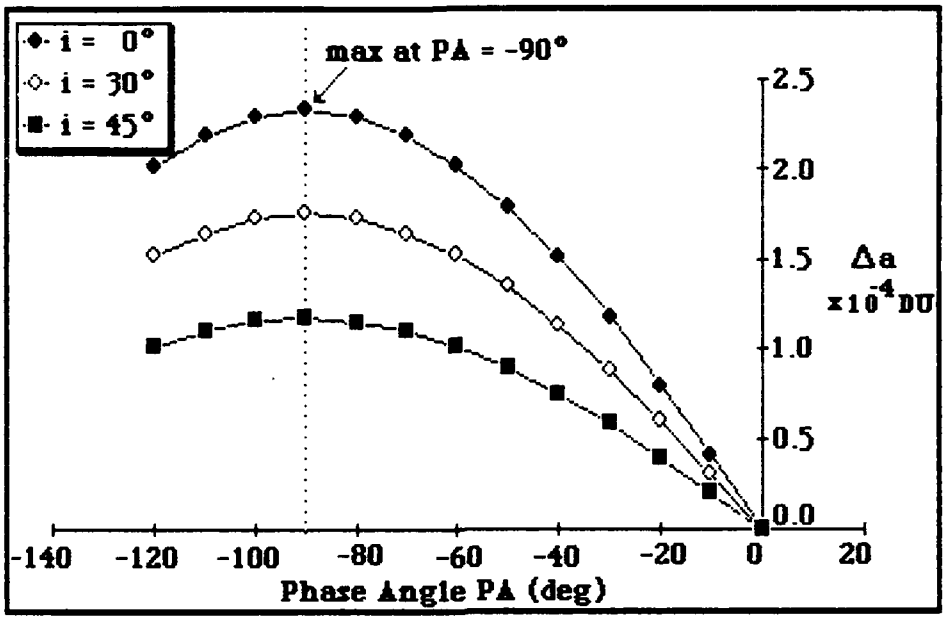


Fig. 4.6: Variation of Angle PA

Interestingly, the minimum occurs when PA = +90 degrees. In all the cases studied in which the IJK frame is congruent to the orbital frame, the following applies:

$$\Delta a \text{ maxima occur at PA} = \phi - \psi = -90^{\circ}$$

$$\Delta a \text{ minima occur at PA} = \phi - \psi = +90^{\circ}$$

This is not the case when two inertial frames are not congruent. As will be seen later, for η and ζ not equal to zero, the maximum location shifts relative to angle PA.

Variation in the Angular Momentum Vector Orientation. The angles η and ζ which describe the angular orientation of the angular momentum vector $\bar{H}$ with respect to the orbital reference frame (ijk) has been superficially neglected. In this study, they have taken on common values of 0° or 90°. Earlier, these angles were termed the "lesser dynamic"" parameters and, therefore, were not varied in the previous perspectives. On the contrary, variations is η and ζ have sufficient control on the behavior of Δa or else they would not have been included. Their major is the shifting the location of the maxima. Here are sample variations:

Fixed η. Figures 4.7A,B,C show just such behavior as η was held fixed at zero and ζ was allow to vary from 30° to 90° for motion in the 30° inclination plane. As ζ is increased, the maxima no longer becomes stationary at PA = -90 but seems to propagate or shift towards PA = 35°.

Fixed ζ. A similar but out of phase situation occurs when ζ is fixed at zero with η allowed to vary. At η = 30°, the Δa function assumes the same shape as in the (η, ζ) = (60,0). This case is depicted by Figures 4.8A,B and C. The (η, ζ) = (90,0) is essentially the tumbling case Jenkins [Ref: 10] investigated earlier.

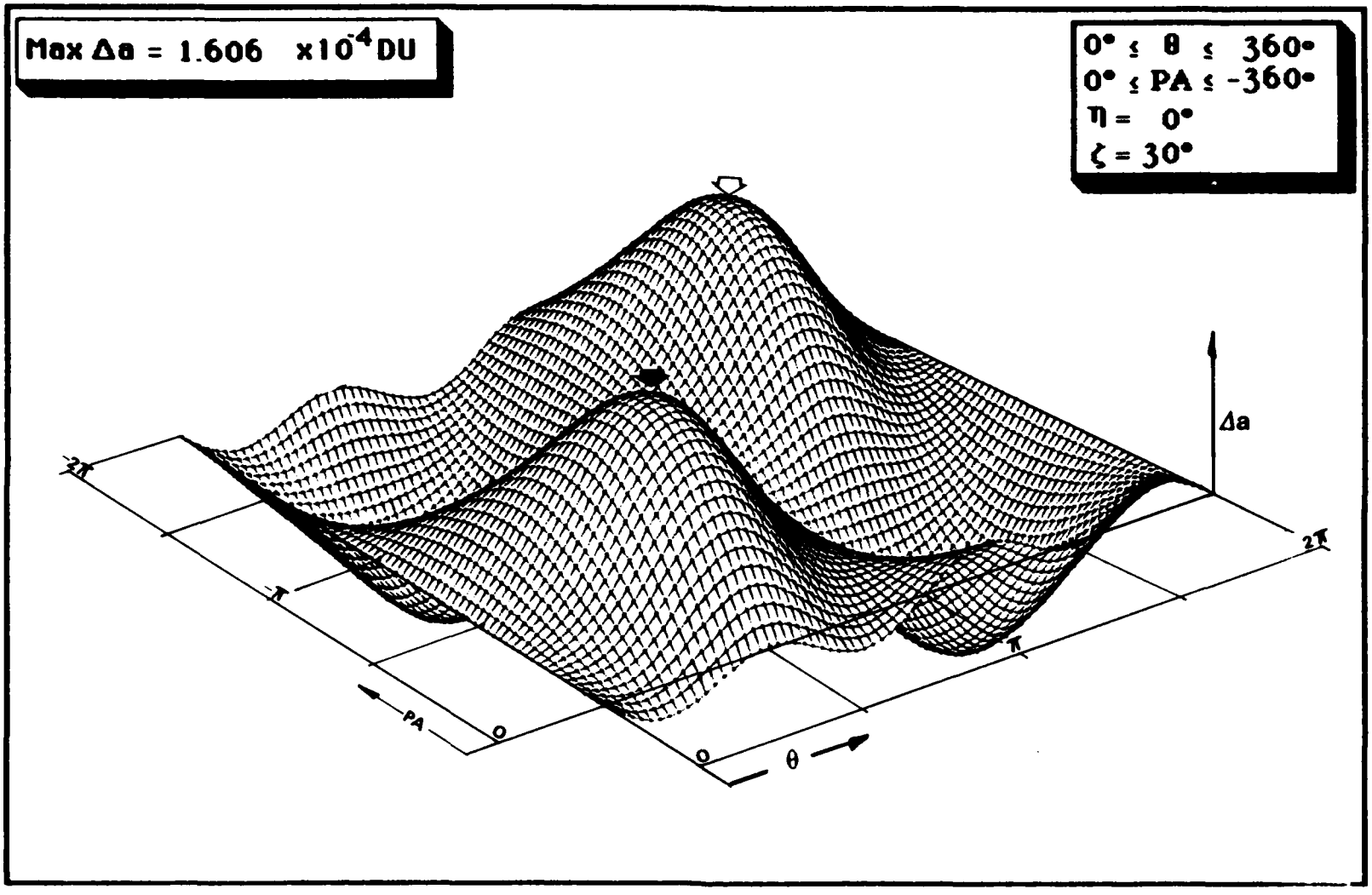


Fig. 4.7A : Perspective (i =30°)

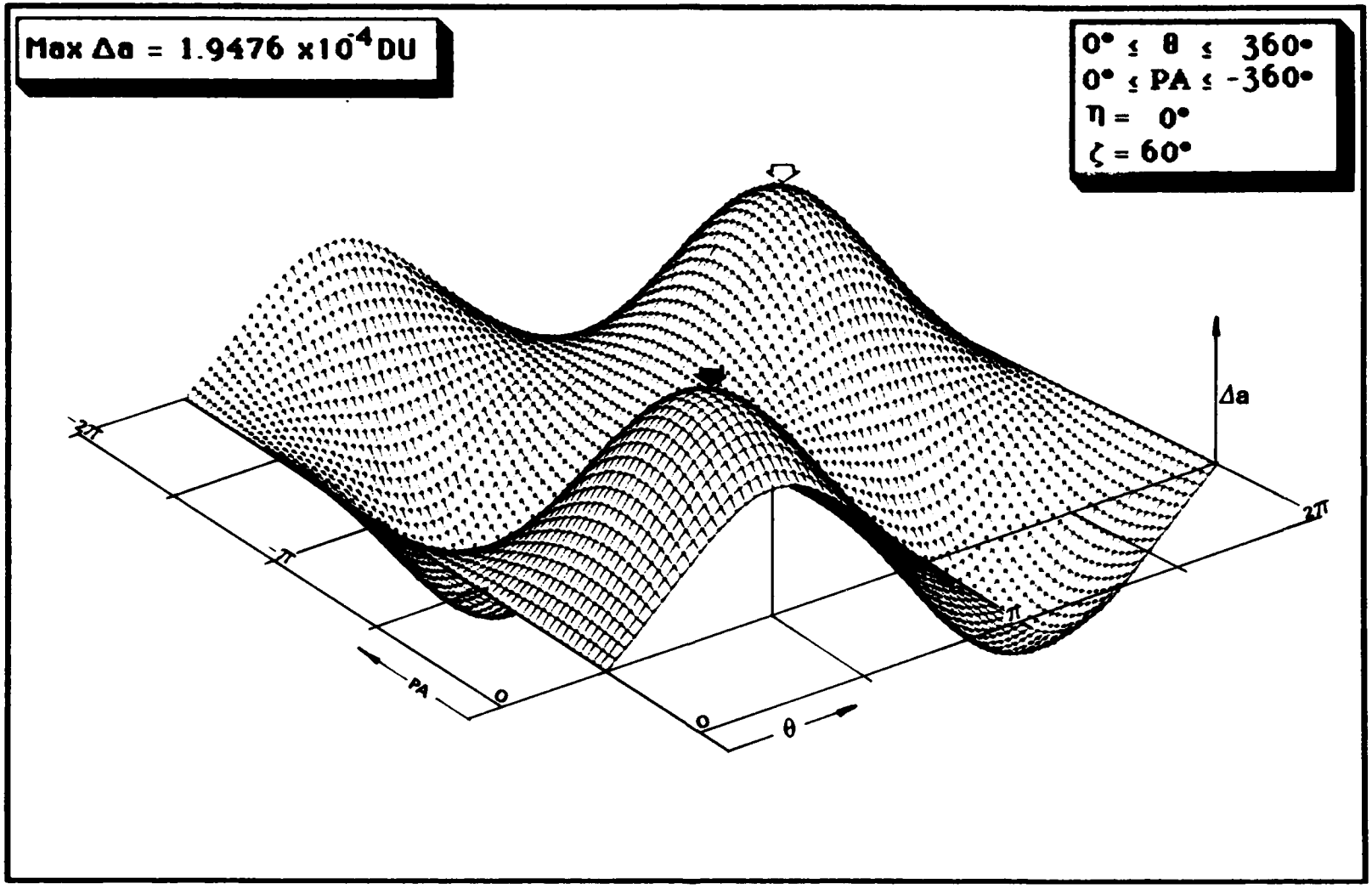


Fig. 4.7B : Perspective (i =30°)

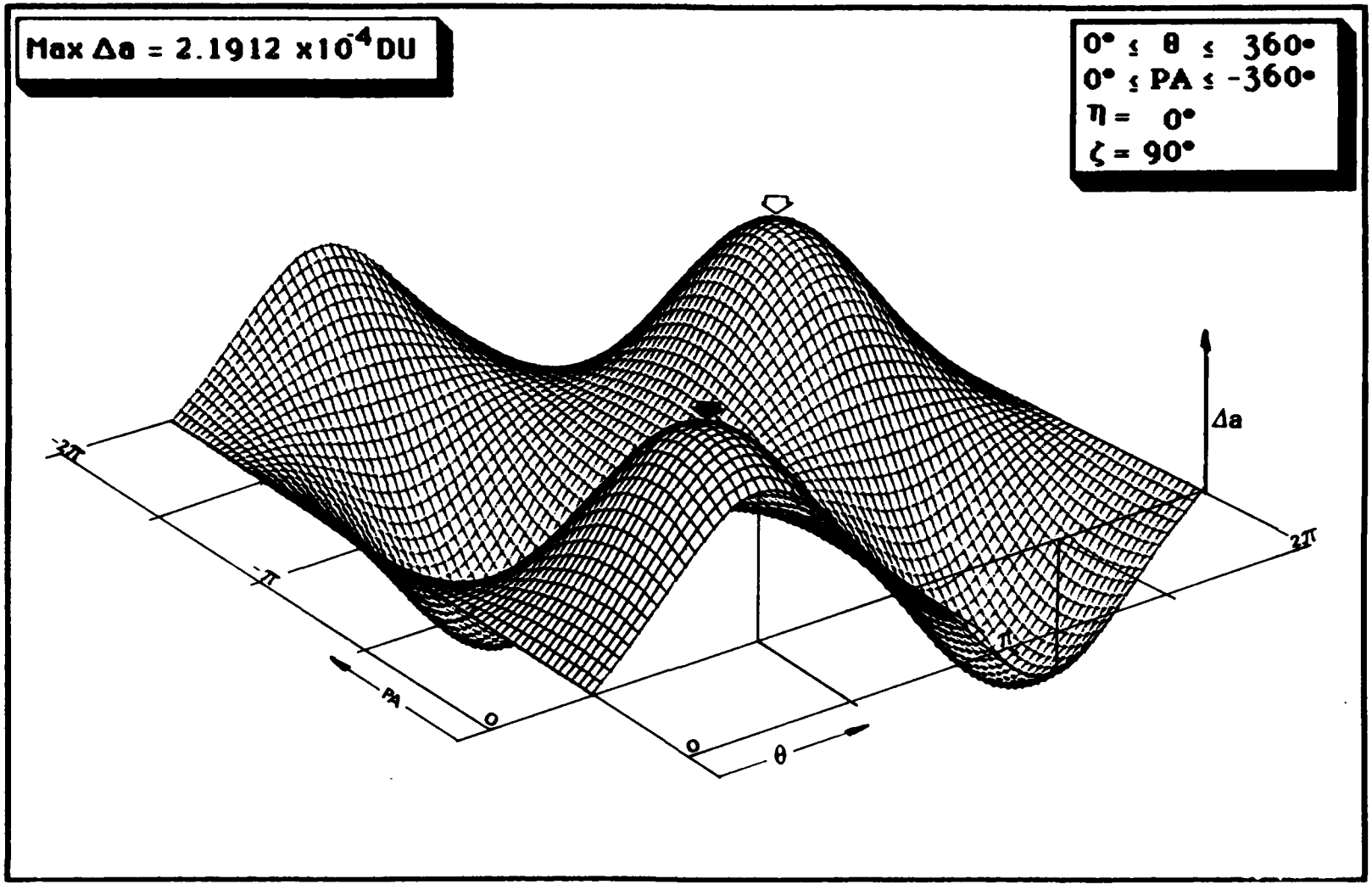


Fig. 4.7C : Perspective (i =30°)

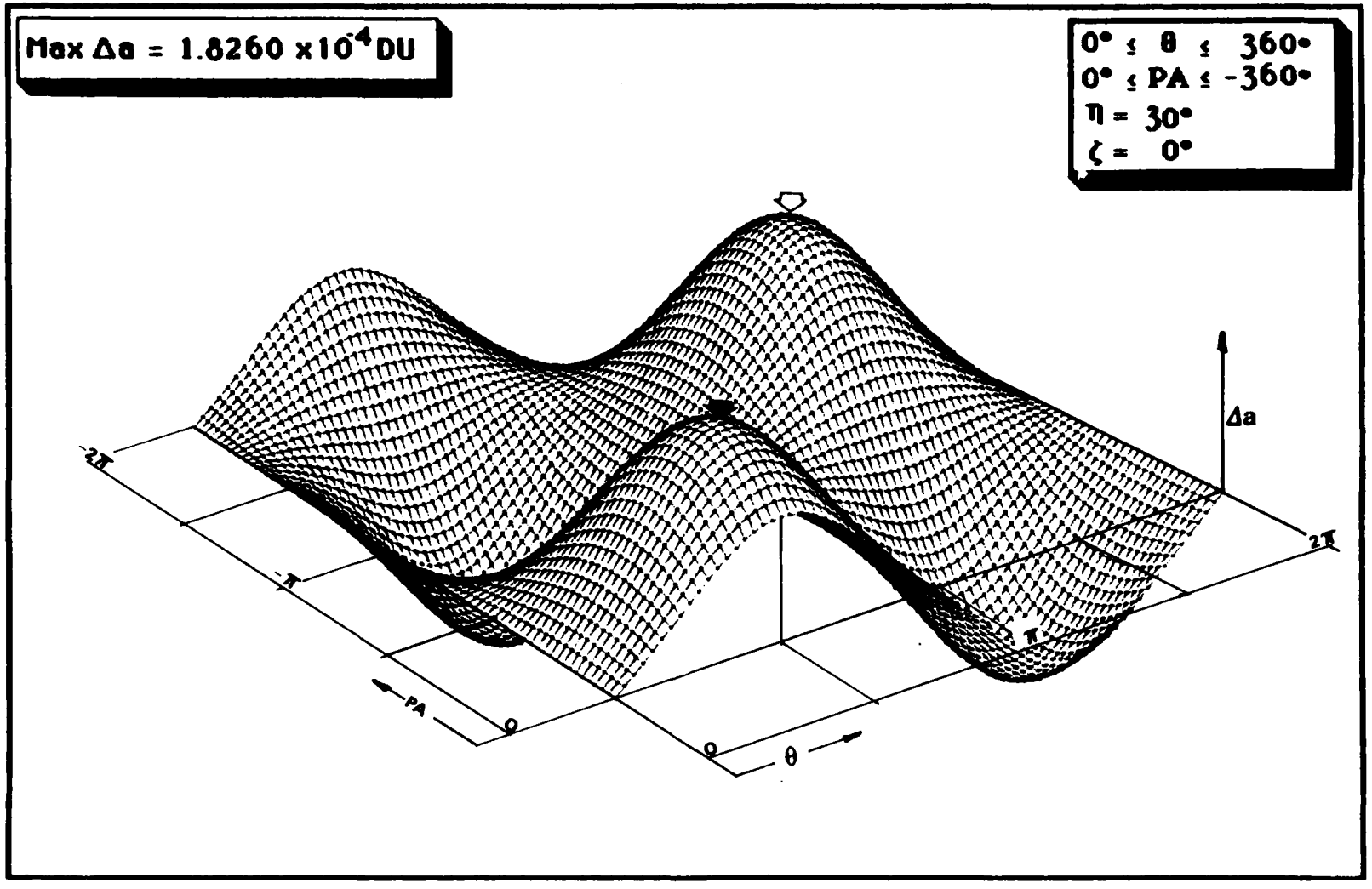


Fig. 4.8.A: Perspective (i =30°)

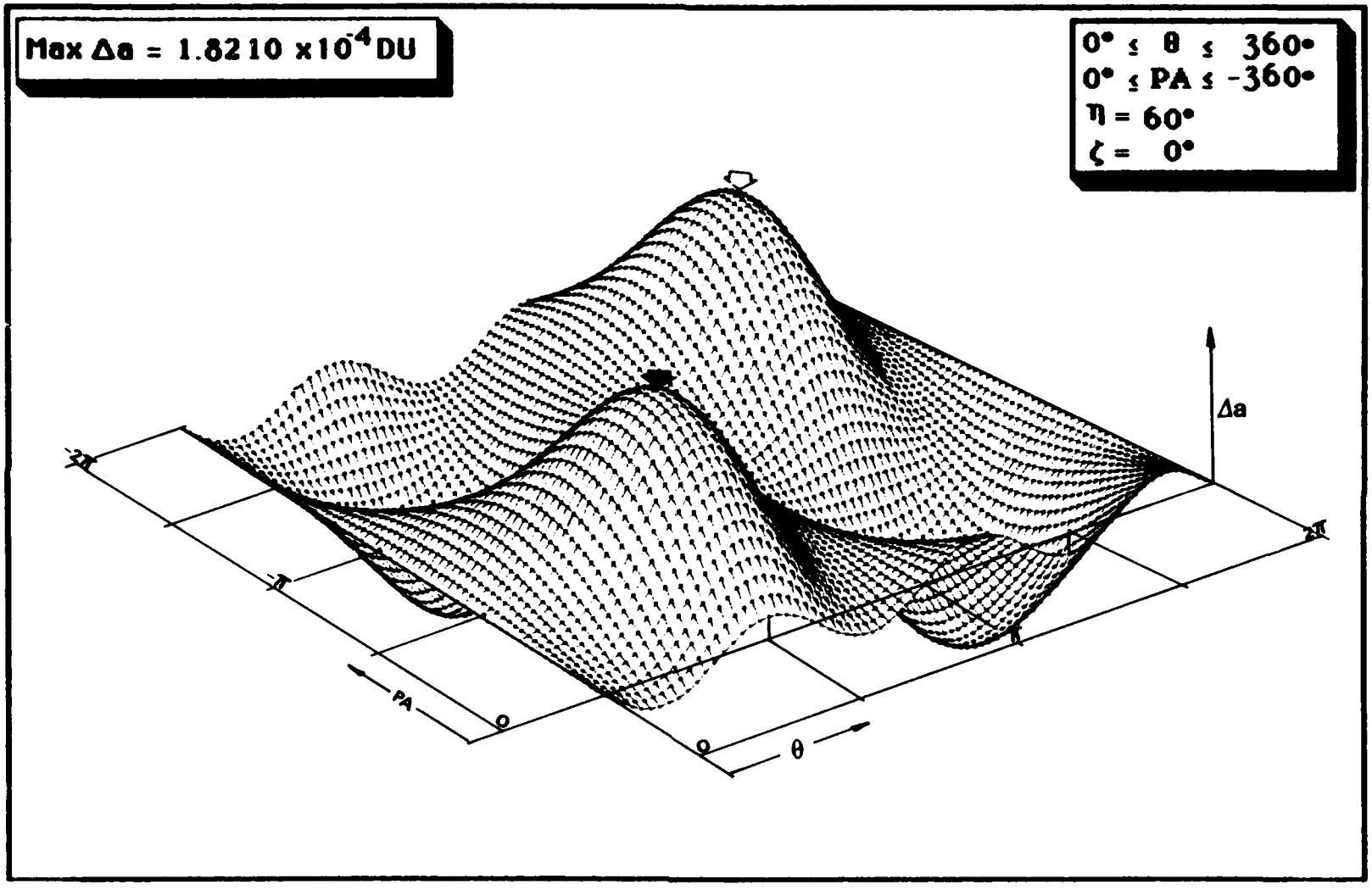


Fig. 4.8.B: Perspective (i =30°)

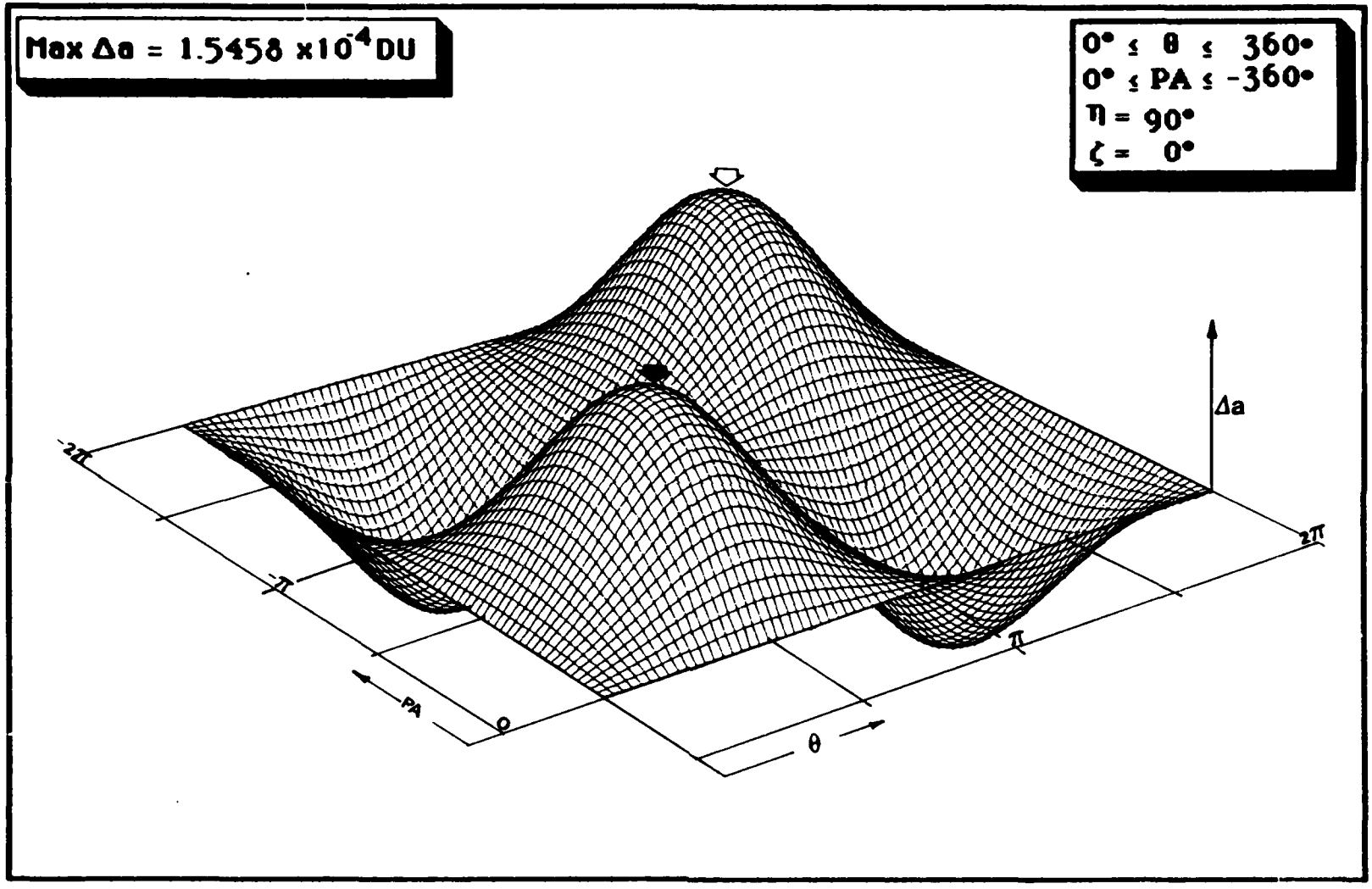


Fig. 4.8.C: Perspective (i =30°)

4.2 Optimization

The search of the optimum set of controls to achieve a maximu change in a per orbit required the use of the perspectioves as a guide in establishing a good intial guess to start up the search algorithm. After noting the location of the maxima from the particular perspective of interest, convergence is almost expected at that graphical coordinates.

Observations. In employing the Stirling Method in search of the required set of optimal control vector, a few peculiar observations were made.. The Stirling Method uses a step size δu to increment through the local area of interest in search for the zero of the function.

a. The search scheme is very sensitive to the step size δu. When the coordinates of the maxima (from perspectives) are inputted, the value of the (C^*) vector components become very small often in the neighborhood of 1×10^{-7}. Since (C^*) is the initial $\partial F_a/\partial u_i$ evaluted at the initial input control values, using near-maxima control values would make the incremental step size too small to differentiate a change that can be discernable by the search scheme. This is where larger step sizes can be used without losing any appreciable accuracy.

b. If the conditions generating the perspectives are used as initial guess inputs, the maxima depicted can be reaccomplished to a higher degree of accuracy. Test cases show this accuracy to be good to the seventh order.

c. The degree of accuracy beyond the seventh order seems to be hampered by round-off or truncation error despite the fact that all computations are accomplished in double-precision mode.

Test Cases. Several test cases were made primarily with the use of the data extracted from Figures 2.4A through 2.4G. Since the intent of the preliminary searches

is to use these figures as the primary feedback for the behavior of the δa function and the verification of the output of the search algorithm. deviations from the initial conditions generating the figures were not made. The results on these preliminary evaluations can be summed up as follows:

1. Searches using coordinates of maxima from the perspectives yielded similar δu required essentially equal to (0.0, 0.0, 0.0, 0.0). This implied that the point of interest is a maxima to begin with and that no changes in the control parameters are required. Convergence is achieved in a single iteration. Further forced iterations resulted in the same; i.e. when search was continued until an iteration limit was reached, the required δu remained at zero. This displayed stability. Larger step sizes did not affect the convergence.

2. Deviations in any single control parameter (holding the other three at initial maxima condition values) resulted in more than one but less than three iterations to converge to a local maxima. From the perspectives, deviations of less than ten degrees seem to be well behaved and predictable in that there are no other local maxima within a 10° radius of any given maxima. Convergence limits of .001 degree and better are realizable.

3. The search algorithm does not provide a check for a global maximum. This handicap and the convergence criterion are responsible for the search converging at a minima adjacent to a maxima. This was a subject of interest in this study.

Results on a test case are presented in Appendix E. The test case selected is for motion in an orbit inclined at 45 degrees. This was of particular interest because of the two adjacent maxima separated by a minimum. This search scheme only searches for the condition in which the first derivative is equal to zero. It is interesting to note that the figure 4.2D (test case, $i=45^{\circ}$), displays 17 different areas at which convergence can be achieved. At any of these areas, the slopes are indeed zero and the search scheme

would these out. Achieving convergence does not signify a maximum Δa. It is apparent from the perspective that there are only four areas in which Δa is maximum. Without reference to the perspective, it would be most difficult to relate convergence values to maxima, minima or inflection points. This makes it so imperative that the search be accomplished hand-in-hand with the corresponding perspective. Only then can the control parameter changes be meaningful.

V. Conclusion

Working in a region of unknowns and searching for an entity yet to be described is just what was initially pursued in this effort. Given a multi-variable function describing the change in the semi-major axis (Δa) with an established objective of finding the maximum change experienced by that function began as a formidable task of high interest and expectation. The pursuit of a solution using known search methods e.g., Stirling Method, to evaluate extrema provided much hope for convergence. Futile preliminary attempts to identify convergence conditions was due to poor initial guesses. This led to question the nature of the Δa function at hand. Understanding what the function does is quite different from knowing what it looks like. The big question is this: What does the Δa function represent graphically? As a resort to a good guess, the four-variable Δa function was graphed perspectively in three-dimensions with two variables held fixed and two varied against the value of the function. The resulting perspectives represent the surface of the function for the given set of "control" conditions. These perspectives give a good overall picture of the behavior of the Δa function. Information that was earlier a guess can now be verified. Regions of diminishing returns and high yield can be identified and search patterns can be concentrated on specific areas. Now a bound exist. Now the initial guess can be nailed down. Search schemes which are sensitive to "good initial guesses" have improved reliability for convergence. The Stirling Method is by no means excluded here. The input of good starting points (u^*) to initiate the search for the optimum control setting that yields the maximum Δa is of critical importance.

Convergence in two or three iterations shows the power of the search method used. Moreover, it shows the goodness of the starting search point. Convergence to coordinates that are known apriori is only possible from reference to the applicable perspectives provided that the starting point is not far from the shown perspective. Deviations far from the local maximum will allow the search mechanism to deviate away from the intended area and converge on another unexpected maximum. This is

normal behavior of such search scheme and there is not need for alarm. The converged control settings can be graphically evaluated by the use of the perspectives defined by the converged control values. As noted earlier, convergence can be achieved at any area satisfying "zero slope" conditions (e.g., maxima, minima, or inflection points). Only by looking at the function's perspectives or by computing its second derivative can the actual maxima desired be identified. The cross-relation between the search scheme results and the perspectives are most necessary for a meaningful output.

Comment on the Equations of Motion

Jenkins [Ref: 10] provided perturbation equations for the one-to-one resonance case. These equations have been thoroughly verified in more than one ways.

1. The equations of motions were integrated for the resonant case (i.e., when the orbital mean motion is equal to the sail precession rate) and found to be correct.

2. When the equations of motion were used to generate the applicable perspectives at a 90° inclination, it was discovered that at a coning angle $\theta = 35°$, a maximum occurs. This is in direct agreement with previous results arrived at by Tsu [Ref: 21] and Garwin [Ref: 5] in which they found that a tilt angle of 35° with respect to the sun provides optimum change in the semi-major axis. The constant coning angle is the tilt angle of the sail with respect to the sun.

These indicate that the perturbation equations arrived at by Jenkins do indeed describe the appropriate motion of a freely coning solar sail.

Lessons Learned

It is fitting that whatever was learned from this academic effort be shared with anyone interested in pursuing a similar quest. The work done here is without its troubles and periods of despair. The biggest difficulty was searching for the maximum of a multi-variable function without any apriori knowledge of its behavior under given conditions. The suggestion here is this: represent the surface (or function) in some kind of perspective (two-dimensional or three-dimensional) and observe its behavior as certain variables are changed while holding others fixed. This would provide valuable insight into what can be expected of the function. Look at it first so that they are no surprises later. This is the biggest lesson learned from this academic effort.

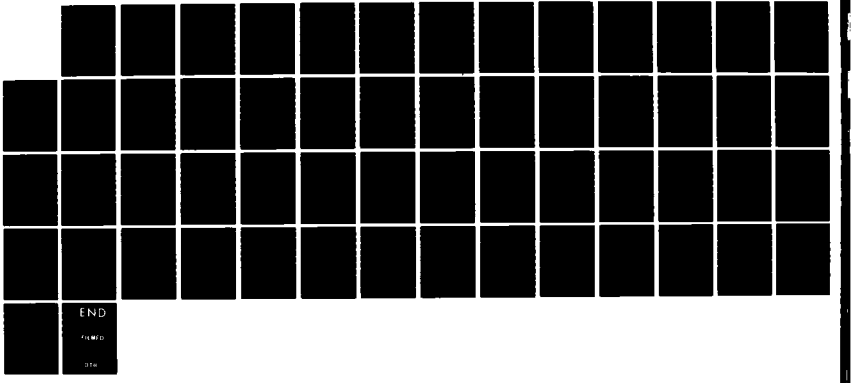
Recommendations

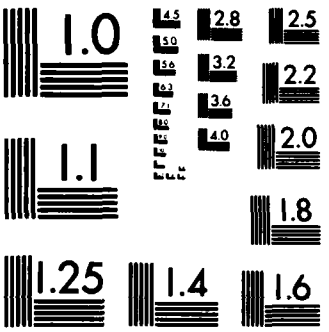
The behavior of this solar sail is just partially known by the exploitation of the Δa function. More can be learned by numerically looking at the behavior of changes in the other state functions eccentricity, inclination and longitude of ascending node. Although, a particular aspect of the inclination is addressed here, it deserves its own segment. The numerical evaluation for a selected performance index can very well be supplemented by producing the three-dimensional perspectives associated with the case of interest.

Another pursuit would be to extend the single-stage dynamic system to a multi-stage dynamic system. This would be the case in finding the optimum series controls which would generate the maximum change in semi-major axis in n revolutions. This would entail applying the single-stage system $n-1$ consecutive times as shown previously in Figure 3.3. Such a multi-stage system can have practical applications.

AD-A153 175 MAXIMIZING THE SEMI-MAJOR AXIS FOR A FREELY CONING SOLAR SAILCRAFT(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 2/2

UNCLASSIFIED M R BORJA DEC 84 AFIT/GA/AA/84D-2 F/G 22/3 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Bibliography

1. Bates, Roger R., Mueller, Donald D. and White, Jerry E., Fundamentals of Astrodynamics. New York, Dover Publications, Inc. 1971.

2. Bryson, A.E.,and Ho, Y.C., Applied Optimal Control. Massachusetts: Ginn and Company, 1969.

3. Cunningham, Capt. John D., "Optimum Contol of a Solar Radiation Pressure Powered Space Vehicle". M.S. Thesis, School of Engineering, AFIT, Wright Patterson Air Force Base, Ohio, December, 1974 (AD-A004-792).

4. Escobal, P.R. and Johnson, R.H., "Eclipse Design Limits for Circular Orbit", The Journal of the Astronautical Sciences, Vol. XIX, 6: 448-461 (May-June, 1972).

5. Fimple, W.R., "General 3-D Trajectory Analysis of a Planetary Escape by Solar Sail", ARS Journal, 32:883-887 (June, 1962).

6. Fixler, Sol Zalel,"Umbra and Penumbra Eclipse Factors for Satellite Orbits" AIAA Journal, Technical Notes, Vol. 2, No 8, August, 1964.

7. Garwin, R. L., "Solar Sailing: A Practical Method of Propulsion within the Solar System," Jet Propulsion, 28: 188-190 (March, 1958).

8. Georgevic, R. M., "Solar Radiation Pressure Force and Torques Model," The Journal of The Astronautical Sciences, Vol. XX, 5: 257-274, (March-April, 1973).

9. Green, Andrew J., "Optimum Escape Trajectory from High Earth Orbit by Use of Solar Radiation Pressure", MS Thesis, School of Engineering AFIT, Wright Patterson Air Force Base, Ohio, August, 1977.

10. Jenkins, K., "Orbital Motion Of a Freely Coning Solar Sail", MS Thesis, AFIT/GA/AA/83D-3, School of Engineering, AFIT, Wright Patterson Air Force Base, Ohio, December, 1983.

11. Kaplan, M.H., Modern Spacecraft Dynamics and Control. New York: John Wiley and Sons, 1976.

12. London, H.S., "Exact Solutions of the Differential Equations of Motion of a Solar Sail with Constant Sail Setting," ARS Journal, 30: 198-200 (February, 1960).

13. Parkinson, R.W. "Effects of Solar Pressure on an Earth Satellite Orbit", Science, Vol. 131, page 920-921, 1957.

14. Polyakhova, Y.N., "Solar Radiation Pressure and the Motion of an Earth Satellit," Journal of Applied Machanics and Technical Physics.

15. Sands, Norman, "Escape From Planetary Gravitational Fields by Use of Solar Sails," ARS Journal, 3133527-531(April, 1961).

16. Schied, Francis, Numerical Analysis Schaum's Outline Series, McGraw-Hill Book Company, New Yoork, 1968.

17. Shames, Irving H., Engineering Mechanics, Vol. II, Dynamics, Prentice Hall, Inc., New Jersey, 1966.

18. Shrivastava, S.K. "Optimum Orbital Control Using Solar Radiation Pressures," AIAA Journal, Vol. 2, No. 8: 1455-1457 (August, 1964)

19. Stoddard, L.G., "Eclipse of Artitical Earth Satellite," Astronuatical Science Review. III, 9-16 (April-June, 1961).

20. Tsander, F.A. "On the Use of Light Pressure Source for Interplanetary Space Travel," 1924-1925. cf. The Problem of Jet-Propelled Flight. Moskva: Mozhplanethye puteshestviya, 1961.

21. Tsu, T.C. "Interplanetary Travel By Solar Sail," ARS Journal, 29: 422-427 (June, 1959).

22. Valeyev, K.G., "Qualitative Investigation of Differential Equation of a Flight by Means of Solar Sail," Journal of Applied Mechanics and Technical Physics, FTD-MT-64-285, Translation. (September, 1965).

23A. Van der Ha, J.C. and Modi, V.J., "On the Maximization of Orbital Momentum and Energy Using Solar Radiation Pressure," The Journal of the Astronautical Sciences, Vol. 27, pp. 63-84, 1979.

23B. Van der Ha, J.C. and Modi, V.J., "Analitical Evaluation of Solar Radiation Induced Orbital Perturbation of Space Structures," Journal of Astronautical Sciences, Vol. XXV, No. 4, pp 2283-306, October-December, 1977.

23C. Van der Ha, J.C. and Modi, V.J., "Orbital Perturbation and Control by Solar Radiation Forces," J. Spacecraft, Vol. 15, No.2, pp 105-112, March-April, 1978.

23D. Van der Ha, J.C. and Modi, V.J., "Solar Pressure Induced Orbital Perturbations and Control of a Satellite in an arbitrary Geocentric Trajectory," AIAA Aerospace Sciences Meeting, 15th, Los Angeles, Ca. 24-26, Paper No. 77-32, January, 1977.

24. Van der Ha, J.C. "The Attainability of the Heavenly Bodies with the Aid of a Solar Sail", Walter-Hohmann-symposium, Paper No. 80-012, European Space Operation Centre,, West Germany, 1980.

25. Wiesel, William, "Problems in Space Flight: published Lecture Notes", Air Force Institute of Technology, School of Engineering, Wright Patterson Air Force Base, Ohio, 1983.

VITA

Mario Reyes Borja was born on 2 June 1949 in Chalan Kanoa, Saipan, Mariana Islands. He graduated from Father Duenas Memorial Seminary & High School, Guam, in 1967. He entered the United States Air Force in September 1968 in which he served as an Avionics Systems Instrumentation Technician. He received his Bachelor of Science in Aerospace Engineering through the Airman's Educational Commisioning Program from the University of Texas at Austin in December 1979. He received his Air Force commission in April 1980 through the Officer Training School, Lackland, Texas. He was then assigned as a Space Propulsion Systems Analyst at the Air Force Rocket Propulsion Laboratory, Edwards AFB, California. He Received his Masters of Science in Systems Management from the University of Southern California through an off-duty extension program in December 1982. He entered the Air Force Institute of Technology, School of Engineering in June 1983 in pursuit of a Masters of Science in Astronautical Engineering.

Permanent Address: 4085 Corolyn Drive
La Mesa, California 92041

Appendix A

Eclipsing Effects

The eclipsing or shadowing of any solar radiation dependent spacecraft is an important phenomenon that must be understood and compensated for in the design stage and/or by some control mechanism of such spacecraft. For this solar sail, the only propulsion source is solar radiation. This makes shadowing a more critical phenomenon for a solar sailcraft than for a spacecraft with a variable-mass propulsion system. To achieve the objective in maximizing select orbital parameters (which are dependent upon the amount of solar radiation incident on the surface), the aspects of shadowing, e.g.,

a. when does shadowing occur, and

b. duration of shadowing,

become particularly interesting to the mission designer in determining the optimum steering controls necessary. The dependence of the changes of the orbital parameters on the shadow time for this particular coning solar sail can be readily seen in the period TP term in Eqn (2.21). Note that TP is the orbital period and is also the time spent in the solar radiation environment during one orbit. Shadowing would result in reducing this TP value and, hence, in adjustments to the amount of perturbational changes the sail's orbit experiences. Just how does one determine a and b above?

Stoddard (Ref: 19) simplifies the aspects of shadowing and presents a means of determining whether or not an artificial satellite in a circular orbit is shadowed and, if it is, what its duration in the shadow. The "circular cylindrical shadow" model is employed instead of a "conical shadow" model which considers the umbra and penumbra shadow components (as presented by Fixler in Ref: 6) to arrive at the following relationships which are referenced in Figures A1 and A2:

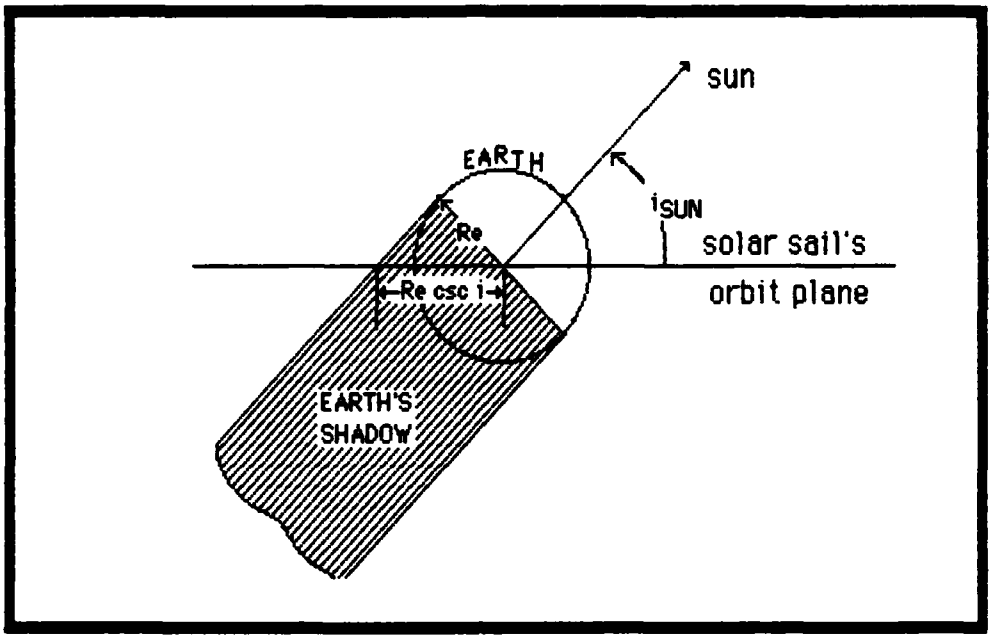


Fig. A1: Earth's Shadow Geometry

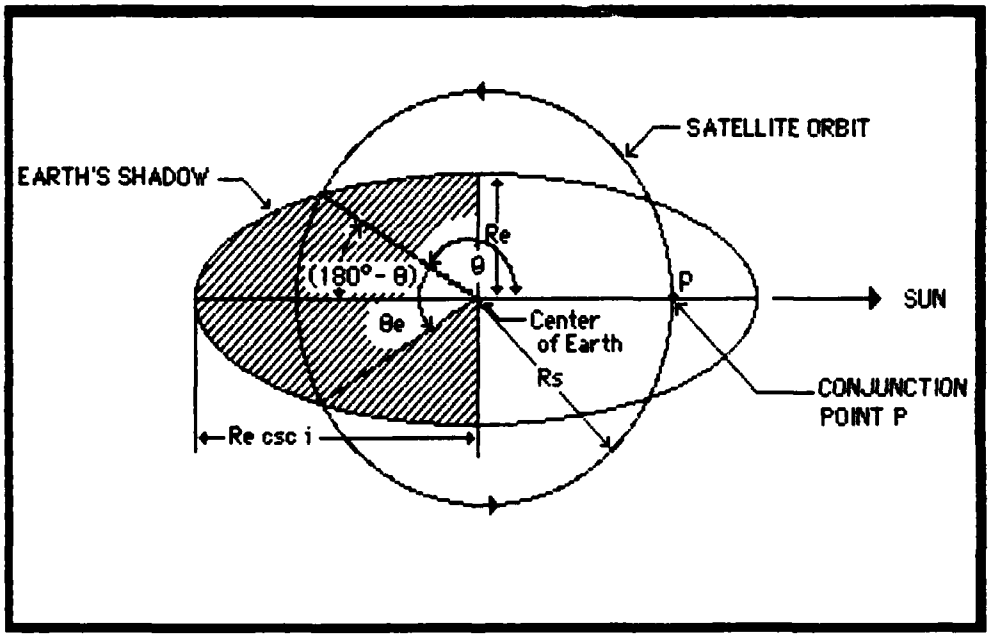


Fig. A2: Projection on Orbit Plane

$$\cos\theta = \pm\ (1/e_{SH})\ \ [1 - (R_E/R_S)^2]^{1/2} \qquad \text{(A-1a)}$$

$$= \pm\ (\sec i_{SUN})\ [1 - (R_E/R_S)^2]^{1/2} \qquad \text{(A-1b)}$$

$$\theta_E = 2(180^\circ - \theta)\,; \qquad \text{(A-2)}$$

$$t_{SH} = (\theta_E/360^\circ)\,TP\,; \qquad \text{(A-3)}$$

where

θ_E ≡ geocentric angle of travel of the satellite during eclipse.

i_{SUN} ≡ geocentric angle between the sun and the satellite's orbit plane.

Note: From Figure A1, this is equivalent to the orbit inclination i.

θ ≡ geocentric angle measured in the orbit plane between the satellite and the conjunction point P (Figure A2).

e_{SH} ≡ eccentricity of the elliptical projection of the earth's shadow on the satellite's orbit plane.

Note: This can be geometrically shown to be equivalent to cos i_{SUN}. This is accomplished in [Ref: 19]

R_E ≡ earth radius - 1 DU.

R_S ≡ satellite orbit radius.

TP ≡ orbital period.

t_{SH} ≡ time the solar sail is in the earth's shadow - duration of shadow.

The limiting case between eclipse or no eclipse is obtained from Equation (A-1) when θ - 180 degrees. Therefore, ...

$$R_S = R_E \csc i_{SUN} = R_E \csc i. \qquad \text{(A-4)}$$

Equation (A-4) is valid for the geometry and the definition of the inclination of the orbit as described in text. That is, if the inclination is defined as the inclination of the orbit plane wrt the ecliptic plane, then this Equation (A-4) will be valid. For any application, reference to the Stoddard article is highly recommended.

From Equation (A-4), eclipsing criteria can be established as follows:

$$\text{if } R_S < R_E \csc i_{SUN} \Longrightarrow \text{a shadow will take place; else} \qquad \text{(A-5)}$$

$$\text{if } R_S > R_E \csc i_{SUN} \Longrightarrow \text{a shadow will } \underline{\text{not}} \text{ take place.} \qquad \text{(A-6)}$$

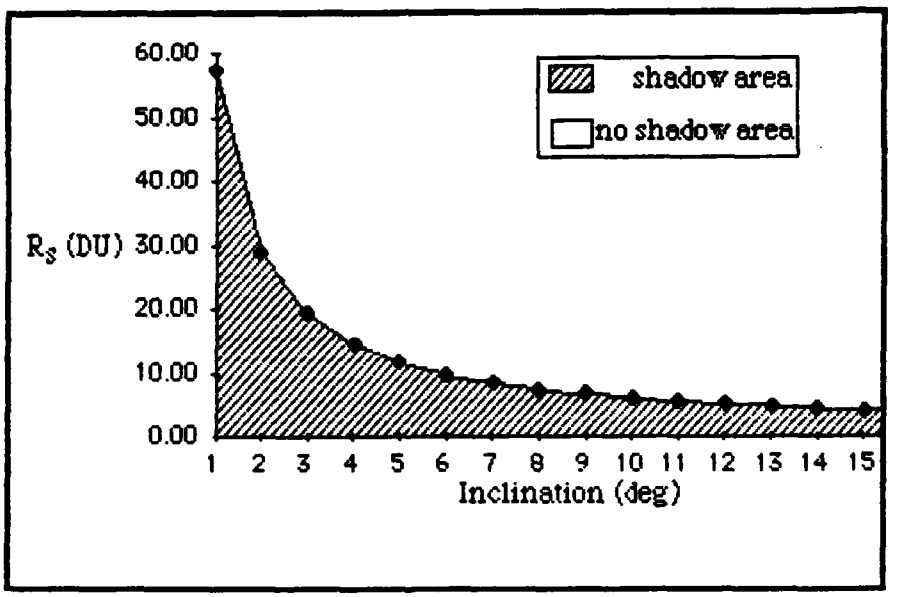


Fig. A3: Solar Sail Shadow Limits

From Equation (A-5), one can determine that any artificial satellite in the ecliptic plane (i = 0°) will experience shadowing. Shadow limits for various inclination i and orbit radius R_S can be seen from the plot of Equation (A-5) in Figure A3 above. The area to the right of the "Limit Line" indicates the region of 'NO SHADOW'.

Table A1 shows the behavior of the various parameters in this relationship as the orbit radius R_S is varied from 2 DU to 10 DU.

Table A1: Shadow Parameters

R_S	Cos θ	θ °	θ_E °	TP(TU)	t_{SH}(TU)	t_{SH}(MIN)	(%)
2.0	**.866**	**150.00**	**60.00**	**17.77**	**2.96**	**39.82**	**11.65**
3.0	.943	160.56	38.88	32.64	3.53	47.47	10.81
4.0	.968	165.47	29.06	50.26	4.04	54.36	8.04
5.0	**.980**	**168.52**	**22.96**	**70.25**	**4.50**	**60.55**	**6.41**
6.0	.986	170.40	19.19	92.34	4.92	66.19	5.32
7.0	.990	171.79	16.43	116.37	5.31	71.40	4.56
8.0	.992	172.82	14.36	142.17	5.67	76.27	3.99
9.0	.994	173.60	12.76	169.64	6.01	80.84	3.54
10.0	.995	174.26	11.48	198.69	6.34	85.19	3.18

For the case in point, the solar sail (at R_S = 2 DU) would have to have an inclination i greater than or equal to 30° to escape any shadowing effects. For solar sail motion in the ecliptic plane, the duration of the shadow t_{SH} is approximately 40 minutes long. This is determined as follows:

$$TP = 2\pi R_S^{1.5} = 2\pi (2)^{1.5} = 17.77 \text{ TU.}$$

$$\theta = \cos^{-1} \left[\pm (\sec 0^\circ)[1 - (1/2)^2]^{1/2} \right] = 150.00^\circ .$$

$$\theta_E = 2(180^\circ - 150^\circ) = 60^\circ .$$

$$t_{SH} = (60^\circ/360^\circ)\ 17.77 \text{ TU} = 2.96 \text{ TU} = 39.82 \text{ min.}$$

These 40 minutes spent in the shadow of the earth equates to 40 minutes less of direct solar radiation exposure. This directly affects the magnitude of the perturbational

changes. From Table A1, one might be alarmed at the increasing magnitude of the shadow time t_{SH} for increasing orbit radius R_S and its resultant decrease in these perturbational changes. The shadow times do indeed increase; but, the percentage increase of shadow time over the period TP decreases as the orbit radius increases. This indicates the relative impact t_{SH} has for high orbiting solar sails. The further out the sailcraft is , the lesser the chances of shadowing become. For the solar sailcraft initiating its maneuver at low-earth-orbit altitudes, this phenomenon is an important issue.

The question is "how much shadowing is tolerable <u>without</u> considering its affects?" The eclipse factor has generally been used to answer this question. The <u>eclipse factor</u> is defined as the ratio of time spent in the sahdow to the orbital period of the satellite. From Eqn (A-3), this can be expressed as

$$\text{Eclipse Factor} = t_{SH}/TP = \theta_E/360^\circ \qquad \text{[Ref: 17]} \qquad \text{(A-8)}$$

If this ratio is small, then shadowing can be neglected. Some type of decision criterion must be formulated to determine an answer to this question. Such criterion will definitely include this eclipse factor and other satellite/mission dependent parameters such thermal constraints, etc.

To get a general conceptual feel for this shadowing effect on a solar sailcraft in the ecliptic plane, a "simply spinning" case with shadowing and another without shadowing were evaluated and contrasted. The data in Figure A4 indicate the decrease in total change in Δa when shadowing effects are considered. There seems to be appreciable relative differences between these computed sets of values, thus indicating the importance of the shadow phenomenon.

Cases dealing in the ecliptic plane are very much affected by this phenomenon. Moving out in altitude and to inclinations greater than zero reduces this effect. This study avoids this phenomenon by having the initial state at a high orbit altitude and a

non-ecliptic inclination such that condition (A6) is satisfied.

Escobal and Johnson [Ref: 3] obtain compact, closed-form expressions for the maximum and minimum eclipse durations of a circular orbit with known semi-major axis and inclination via simple geometric constructions. The method outline provides the designer with the tool to evaluate the envelope of eclipse durations a spacecraft would experience throughout its lifetime.

Polyakhova [Ref: 14] provides a more extensive and thorough coverage of the shadowing phenomenon. This study develops the solar constant equation from the basic quantum theory of light. It further investigates the shadow effects in the case of the radiation-pressure influence on the secular acceleration of the satellite, i.e., on the quantity $\Delta TP/TP$ (the variation of the satellite period during one orbit). It addresses orbits of arbitrary eccentricity and develops a shadow equation and provides solution for certain simplifying conditions. The author develops the equation using non-standard reference frames. For this reason, the development is not pursued here. Strong recommendations is made to it for those pursuing a more rigorous approach to this shadow phenomenon.

Appendix B

Forming the Sensitivity Matrix [C]

This following exercise is presented to bring awareness to the extend of labor one must suffer in determining the partials of the change in the semi-major axis Δa to incremental changes in the control parameters (θ, η, ζ, PA). Recall, PA $\equiv$ $(\phi-\psi)$. This exercise is as follows:

$$[C] = \left[\partial \Delta a/\partial\theta \quad \partial \Delta a/\partial\eta \quad \partial \Delta a/\partial\zeta \quad \partial \Delta a/\partial PA \right] \tag{B-1}$$

where (from Equation (2.21),

$$\Delta a = a^{1.5} D\, TP/\mu \left[da_1 + da_2 + da_3 + da_4 + da_5 + da_6 \right] \tag{B-2}$$

Using the chain rule for differentiation and looking at the first element in Equation (B-1),

$$\frac{\partial \Delta a}{\partial\theta} = \left[\frac{a^{1.5} D\, TP}{\mu}\right]\left[\frac{\partial da_1}{\partial\theta} + \frac{\partial da_2}{\partial\theta} + \frac{\partial da_3}{\partial\theta} + \frac{\partial da_4}{\partial\theta} + \frac{\partial da_5}{\partial\theta} + \frac{\partial da_6}{\partial\theta}\right] \tag{B-3}$$

Working solely with the first term da_1 on the rhs of Equation (B-3),

$$da_1 = \tfrac{1}{4}(d_1)^2 \left\{(3\beta_2 + \beta_4)\cos PA - (3\beta_1 + \beta_5)\sin PA\right\} . \tag{B-4}$$

$$\partial da_1/\partial\theta = \partial(d_1)^2/\partial\theta \left\{(3\beta_2 + \beta_4)\cos PA - (3\beta_1 + \beta_5)\sin PA\right\}$$

$$+ (d_1)^2\left\{3\partial\beta_2/\partial\theta + \partial\beta_4/\partial\theta\right\}\cos PA$$

$$- (d_1)^2\left\{3\partial\beta_1/\partial\theta + \partial\beta_5/\partial\theta\right\}\sin PA \qquad \text{(B-5)}$$

Expanding the d_1 and $\partial\beta$'s terms above, where...

$$d_1 = \sin\theta\cos\zeta\cos i, \qquad \text{(B-6)}$$

and

$$\beta_1 = \sin\theta\cos\zeta, \qquad \text{(B-7a)}$$

$$\beta_2 = \sin\theta\sin\zeta\cos\eta, \qquad \text{(B-7b)}$$

$$\beta_4 = \sin\theta\sin\zeta, \qquad \text{(B-7c)}$$

$$\beta_5 = \sin\theta\cos\eta, \qquad \text{(B-7d)}$$

one arrives at the following results:

$$\partial da_1/\partial\theta = 1/8\left[2s2\theta\, c^2\zeta\, c^2 i\left\{(3s\theta s\zeta c\eta + s\theta s\zeta)cPA - (3s\theta c\zeta - s\theta c\eta)sPA\right\}\right.$$

$$\left. + (1 - c2\theta)\left\{(3c\theta s\zeta c\eta + c\theta s\zeta)cPA - (3c\theta c\zeta + c\theta c\eta)sPA\right\}\right] \qquad \text{(B-8)}$$

where c ≡ cos and s ≡ sin.

Note that Equation (B-8) takes care of the first term in Equation (B-3). There are several more terms that require similar expansions and differentiations wrt the control parameter θ. Specifically,

$$\partial da_2/\partial\theta = \underline{?} \qquad \partial da_3/\partial\theta = \underline{?}$$

Extending this to the partials of the remaining control parameters (η, ζ, and PA), one can easily imagine the amount of labor required and the countless room for error(s).

$$\partial\Delta a / \partial\eta = \underline{?} \qquad \partial\Delta a / \partial\zeta = \underline{?} \qquad \partial\Delta a / \partial PA = \underline{?}$$

This study circumvents this potentially troublesome area and approaches the determination of the Sensitivity Matrix C by employing numerical techniques. This approach is discussed in the text.

Appendix C

The Surface Program. The Fortran Program constructed to generate the various 3-dimensional perspectives of Δa is included here in its entirety. It is designed to be user friendly; it will lead any prospective user to easily generate similar perspectives provided that there is some basic understanding of the mechanics of the UNIX/VAX systems and the operation of the HP7220v Plotting Table.

Generating the Data. Equations (2.21) , (2.22), (2.29) and (2.30) were programmed in a user-interactive program called ZDATA to generate the changes in the semi-major axis parameter. Desired inputs queried include the following:

a. the initial semi-major axis

b. the orbit inclination

c. the angular momentum orientation angles (η and ζ)

d. the range in the coning angle (θ), and

e. the range in the phase difference angle (PA = ϕ-ψ).

Additional inputs in scale factor are required to arrive at a data value that can be easily unterpreted. Scalex increments the ranges of the angles θ and PA to allow a macro-perspective or a micro-perspective of the a function. Scalez raises the data to a value between -10.0 and 10.0 for easy interpretation of relative magnitudes of Δa. The actual Δa is preserved in the parameter called z .

The information generated is placed into a data file (called plotdata) compatible to the plotting routine. The Program Listing is attached.

The Plotting Routine. This is basically the set of instructions for getting the data plotted on an HP7220v Plotting Table. Additional information can be obtained from the S-Package routine. The instructions are as follows:

Instructions

1. Locate the terminal co-located with the HP7220v Ploting Table.
2. Login as usual and move into the directory in which the plot data is located.
3. On the terminal, enter the following:
   a. set term=h19 (this connects terminal to UIX/VAX system)
   b. hit SETUP key.
   c. hit B key and set baud rate to 2400; verify similar setting on rear of HP7220v device.
   d. hit SETUP key to lock value in.
4. Turn plotter on and set peper & pens up; place plotter on STBY
5. On the terminal, enter the following (*note*: % and < are computer prompts)
   a. % S (response will be <)
   b. < hp7220v (this identifies ploting device)
   c. > z_matrix(read("data filename"),nrow=xx,byrow=T)
      where filename = filename used in generating data
      nrow = range x scalez + 1
      (see S-Package reference for additional options)
   d. z_ z-min(z)
   e. z_z/max(z)
   f. persp(z) (see S-Package reference for additional options)

Note: 5d and 5e dimension the data to the maximum value of z contained in data set. Relative magnitudes can be expressed by inserting the maximum value of Δa obtainable in step 5d above. The resulting perspective will give the relative magnitudes of the a values wrt to this maximum.

Caution: The underscore _ following the term z in the instruction above is an important function of the plotting routine and must not be neglected.

Program Listing

```
c-------------------------------------------------------------------
      PROGRAM ZDATA
c
comment:  This program generates the plotdata for input into the
c     S-Package routine.  The data is used in program persp(z).
c     modified for  pa vs theta  (pa is horizontal)
c     surface generation routine (2 variable; 3-dim)
c     Double precision is not required for perspectives.
c
      integer i,j,np1,np2,scalex
      integer dif1,dif2,i1,i2,f1,f2,dec1,iter
      integer dec3,dec4,icase
      real da(0:100,0:100),dax(0:100,0:100),scalez,datot,coeff
      real a,ix,in,tp,sac,conv,pi
      real theta,eta,etax,chi,chix,pa
      real b1,b2,b3,b4,b5,b6,b7,b8,d1,d2,d3
      real da1,da2,da3,da4,da5,da6
c
      data sac/.00000465/
c-------------------------------------------------------------------
comment:  open up tapes file for output.  Plot data contains the
c         all the necesarry data to generate the perspectives.
c         No comments are allowed in this data set. Plotinfo con-
c         tains all pertinent information to plotdata.
c
c     open(unit=11,file='plotdata',access='sequential',
     $ status='new')
      open(unit=12,file='plotinfo',access='sequential',
     $ status='new')
        rewind(unit=11)
        rewind(unit=12)
c-------------------------------------------------------------------
c
comment:  This is the USER INTERACTIVE portion.  It generates on c
          screen prompts for required input data.
```

```
c
10    print*,'Enter a,i,eta,chi    (canonical & degrees) (real)'
      read*,a,ix,etax,chix
      print*,'CAUTION: Do not exceed a 100 X 100 array!'
      print*,'Enter SCALEX   (e.g., 5 pt*deg = 5)  (integer)'
      read*,scalex
20    print*,'Enter INITIAL & FINAL points for THETA   (integer)'
      read*,i1,f1
      print*,'Enter INITIAL & FINAL points for PA      (integer)'
      read*,i2,f2
      print*,'Enter SCALEZ  (desired height for DA)   (real)'
      print*,'note: scalez = 10000. (das right! 10 K) works'
      read*,scalez
      print*,'Are you sure, Solar Sailor?     1=YES;  0=NO'
      read*,dec1
      if(dec1.ne.1) go to 10
c-------------------------------------------------------------------
comment:  Compute some important parameters.
c
      pi = 4.*atan(1.)
      tp = 2.*pi*sqrt(a*a*a)
      conv = pi/180.
      in = ix*conv
      chi = chix*conv
      eta = etax*conv
c
comment:  echo input back to screen.
c
          print*,'period tp   (tu)  =',tp
          print*,'inclination (deg) =',ix
          print*,'eta         (deg) =',etax
          print*,'chi         (deg) =',chix
c
```

```
comment:  Determine the intervals for plot. Intervals must be
c          equally spaced.
c
      dif1 = iabs(f1 - i1)
      dif2 = iabs(f2 - i2)
      np1 = dif1/scalex
      np2 = dif2/scalex
c
      print*,'dif1 = ',dif1,'     dif2 = ',dif2
      print*,'np1  = ',np1,'      np2  = ',np2
      iter = 0
c
      write(12,190)a,ix,etax,chix,i1,f1,i2,f2,scalex,scalez
c
      write(12,180)
      write(*,180)
 180  format(//30x,'PA',/)
c-------------------------------------------------------------------
c*************************** DO LOOP ****************************
c
c     np1 is the # of points in Theta parameter.
c     np2 is the # of points in PA parameter.
c     scalex is the # of divisions desired.
c     scalez is the factor required to raise the Da value to some
c             value that can be worked with.
c
      do 100 i = 0,np1
      theta = (i1 + i*scalex)*conv
c     print*,'     theta = ',theta
            do 110 j = 0,np2
            pa= (i2 - j*scalex)*conv
c           print*,'     pa    = ',pa
c
```

```
comment:  Compute the various parameters in the Da equation.
c
            b1 = sin(theta)*cos(chi)
            b2 = sin(theta)*sin(chi)*cos(eta)
            b3 = sin(theta)*sin(chi)*cos(eta)
            b4 = sin(theta)*sin(chi)
            b5 = sin(theta)*cos(eta)
            b6 = cos(theta)*cos(chi)*sin(eta)
            b7 = sin(theta)*sin(eta)
            b8 = cos(theta)*cos(eta)
c-----------------------------------------------------------------
            d1 = sin(theta)*cos(chi)*cos(in)
c
            d2 = sin(theta)*sin(chi)*cos(eta)*cos(in)
     $           + sin(theta)*sin(eta)*sin(in)
c
            d3 = cos(theta)*sin(chi)*sin(eta)*cos(in)
     $              - cos(theta)*cos(eta)*sin(in)
c-----------------------------------------------------------------
      da1 =  .25*d1**2.*((b2+3.*b4)*cos(pa)-(3.*b1+b5)*sin(pa))

      da2 = -.50*d1*d2*((b2-b4)*sin(pa)+(b5-b1)*cos(pa))
c
      da3 = 2.00*d1*d3*(b6*cos(pa)-b3*sin(pa))
c
      da4 = 2.00*d2*d3*(b3*cos(pa)+b6*sin(pa))
c
      da5 =  .25*d2**2.*((3.*b2+b4)*cos(pa)-(b1+3.*b5)*sin(pa))
c
      da6 = d3**2.*((b2+b4)*cos(pa)-(b1+b5)*sin(pa))
c-----------------------------------------------------------------
      datot = da1+da2+da3+da4+da5+da6
      coeff = a**1.5*sac*tp
      dax(i,j) = coeff*datot
      da(i,j)  = scalez*dax(i,j)
      iter = iter + 1
 110  continue
```

```
        write(11,200)(da(i,j),j=0,np2)
        write(12,200)(da(i,j),j=0,np2)
        write(*,200)(da(i,j),j=0,np2)
 100  continue
c
c_____________________________ FORMATS _____________________________
 170  format(/5x,'tp =',f6.3,2x,'in =',f6.3,2x,'eta =',f6.3,'chi
     $      =',f6.3)
c
 190   format(//,'a =',f6.3,2x,'in =',f6.3,2x,'eta =',f6.3,2x,'chi
    $ =',f6.3, 2x,'theta from:',i4,' to',i4,2x,'pa from:
     $      ',i4,' to',i4,/,'scalex =',f3.1,2x,'scalez =',f10.2)
c
 200     format(100(f7.3))
c_____________________________________________________________________
      print*,'Do you want to try another case????  1=YES; 0=NO'
                read*,icase
            if(icase.ne.0) then
            print*,'a =',a
            print*,'i =',ix
            print*,'eta =',etax
            print*,'chi =',chix
            print*,'Any change in a,i,eta,chi?      1=YES; 0=NO'
                read*,dec3
            if(dec3.ne.0) go to 10
c
                print*,'theta starts at',i1,' and ends at',f1
                print*,'pa starts at',i2,'  and ends at',f2
                print*,'Any change in theta and pa? 1=YES; 0=NO'
                read*,dec4
                if(dec4.ne.0) go to 20
            endif
c
        endfile(unit=11)
        endfile(unit=12)
        stop
        end
```

Appendix D

Optimization Program

Contents. The software constructed to determine the set of control vectors that would make the performance index J stationary consisted of the following.

1. Main Program: ZERO F
2. Subroutine: ECHO
3. Subroutine: UPART
4. Subroutine: DELTA
5. Subroutine: INV4X4

Description. Here is a short synopsis of the function of each program used in this search scheme. The entire listing is documented further in each section of the main where additional information would facilitate its understanding. The programs are as follows:

Program ZERO.F. This is the main driving program that employs the above subroutines to develop the necessary first and second partials and to determine the zero of the optimality condition. It is a user friendly program written to work interactively. It queries the user for necessary input data, an iteration limit, and a convergence limit.

Subroutine ECHO. The input data along with generated preliminary information (e.g. orbital period) are echoed back to the user for verification.

Subroutine UPART. The partials of the state function Fa with respect to the control variables ($\theta, \eta, \zeta, \phi$-$\psi$) are computed via the Stirling (a.k.a modified Newton-Rhapson or central differencing) iteration method. This program is summoned four times to construct the optimality control vector and supporting partials [PC]. Output is $\partial^2 F_a/(\partial u_i u_j)$ in the form [PUA] and [PUB].

Subroutine DELTA. This uses the perturbation equations to determine the changes in the orbital parameters a. Output is Δa.

Subroutine INV4X4. This is used to invert the 4X4 [PC] matrix so that the δu required can be determined. It outputs the determinant and inverse of [PC] - [PCI].

Program Listing. Attached is the entire program listing of the main program ZERO.F. All supporting subroutines required to compile and execute the main program are included here; it is a stand-alone program.

Optimization Program Listing.

The following are the computer codes written to search for the stationary value of the Performance Index J. The **Stirling Method** is used to iterate and find the zero of the function that would allow this stationary value to exist. Equations found in Section 3 under DISCUSSION are the basis for this program.

```
      PROGRAM ZERO
C
C     Last modified on 25  Nov 84      0600 hrs
C
C    This program drives the partial routines to determine the zero
C         of the optimality condition:
C         Lamda (Transpose)*Upartials = [0]
C
      INTEGER DEC0,DEC1,DEC3,DEC5,DEC6,DEC7,DECLAS
      INTEGER DEC8,DEC10,DEC13,DEC14,DEC50,DATE
      INTEGER iterx,ii,jj,m,n,part,k,iter,case
C
      DOUBLE PRECISION PU(4,4),PC(4,4),PCI(4,4),DIFF(4,4)
      DOUBLE PRECISION del(4),deldeg(4),C(4),PUA(4,4),PUB(4,4)
      DOUBLE PRECISION a,i,ax,ix,cv,det,tp,sac,mu,dap,dam,da
      DOUBLE PRECISION theta,eta,chi,pa,thetax,etax,chix,pax
      DOUBLE PRECISION conv,dconv,tconv,pi,tpi
      DOUBLE PRECISION dthe,deta,dchi,dpa,delu,du,dux
C
      DATA ax,ix/2.d+00,30.d+00/
      DATA thetax,etax,chix,pax/90.d+00,0.d+00,0.d+00,-90.d+00/
      DATA SAC,iter/4.65d-06,0/
      DATA DCONV/7.9053682d+00/
      DATA TCONV/806.81118742d+00/
      DATA mu,cv/1.d+00,.001d+00/
      DATA dap,dam/0.d+00,0.d+00/
      DATA dthe,deta/0.d+00,0.d+00/
      DATA dchi,dpa/0.d+00,0.d+00/
      DATA delu,dux/.01d+00,.05d+00/
```

```
C
COMMENT:  Open up a tapes for print files to be stored in.
C         unit 19 contains the del(ii) results with sys description.
        open(unit=19,file='output19',access='sequential',
     $     status='new')
        rewind(unit=19)
C_____________________________________________________________________
COMMENT:  This starts the interactive mode for data input purposes.
C
        print*,'Enter DATE and CASE NO  (e.g. 180984,5):'
        read*,date,case
 204    print*,'Any changes in ORBITAL PARAMETERS?  1=Y;  0=N'
        read*, DEC0
        if(DEC0.ne.0) then
          print*,'Semi-major Axis a =',ax    ,'Any Change? 1=Y; 0=N'
          read*,DEC1
          if(DEC1.ne.0) then
            print*,'ENTER a  (DU)      (double precision)'
            read*,ax
          endif
          print*,'Inclination i =',ix        ,'Any Change? 1=Y; 0=N'
          read*,DEC3
          if(DEC3.ne.0) then
            print*,'ENTER i   (deg)     (double precision)'
            read*,ix
          endif
        endif
C_____________________
 208    continue
C_____________________
        print*,'Current theta =',thetax,'    ANY CHANGE?  1=Y; 0=N'
        read*,DEC5
        if(DEC5.ne.0) then
          print*,'Enter theta  (deg)  (double precision)'
          read*,thetax
        endif
        print*,'Current eta =',etax,'        ANY CHANGE?  1=Y;0=N'
        read*, DEC6
```

```
 if(DEC6.ne.0) then
   print*,'Enter eta   (deg)   (double precision)'
   read*,etax
 endif
 print*,'Current chi =',chix,'        ANY CHANGE?  1=Y;0=N'
 read*, DEC7
 if(DEC7.ne.0) then
   print*,'Enter chi   (deg)   (double precision)'
   read*,chix
 endif
 print*,'Current pa =',pax,'          ANY CHANGE?  1=Y;0=N'
 read*, DEC8
 if(DEC8.ne.0) then
   print*,'Enter pa  (deg)   (double precision)'
   read*,pax
 endif
print*,'Current delu =',delu,'        ANY CHANGE? 1=Y ;0=N'
read*,dec10
if(dec10.ne.0) then
  Print*,'Enter delu        (double precision)'
  read*,delu
endif
print*,'Current du =',dux,'           ANY CHANGE? 1=Y; 0=N'
read*,dec13
if(dec13.ne.0) then
  print*,'Enter du    (double precision)'
  read*,dux
endif
print*,'Do you want to iterate?                    1=Y; 0=N'
read*,dec14
 if(dec14.ne.0) then
   print*,'Enter No. of Iteration  (integer format)'
   read*,iterx
   print*,'Convergence Limit =',cv,'   ANY CHANGE? 1=Y;,0=N'
   read*,dec50
     if(dec50.ne.0) then
     print*,'Enter Convergence Limit -v  (double precision)'
     read*,cv
```

```
         endif
       endif
       print*,''
       print*,'ARE YOU SURE OF ALL CHANGES?     1=Y; 0=N'
       print*,''
       read*, DECLAS
         if(DECLAS.EQ.0) GO TO 204
C_____________________________________________________________________
COMMENT:  Initialize the system state parameter:
          a  = ax
C
COMMENT: CALCULATE PI IN A UNIQUE WAY.
C
       PI   = 4.d+00*DATAN(1.d+00)
       TPI  = 2.d+00*PI
       CONV = PI/180.d+00
          i = ix*conv
C
COMMENT:  Calc the period of the initial system w/o perturbation.
C
        TP = TPI*DSQRT(A**3.d+00/MU**2.)
C_____________________
 209    continue
c_____________________
comment:  If iteration is desired, 209 will bring in the Unew values
c          that were computed from UPART. The control parameters are
c          updated so that they are stepped through the function
c          until convergence is met or until an  iteration limit
c          is reached.  Recall,
c                Delta u = Unew - Uold
c                Unew = Uold + Delat u
c_____________________________________________________________________



COMMENT:  PRINT HEADER FOR OUTPUT:
C_____________________________
       if(I.EQ.0.d+00) then
```

```
          write(*,100) date,case,iter
          write(18,100) date,case,iter
          write(19,100) date,case,iter
          else
          write(*,101) date,case,iter
          write(18,101) date,case,iter
          write(19,101) date,case,iter
        endif
C_____________________________________
COMMENT:  Echo back the initial system:
c_____________________________________________________________________
          CALL ECHO(Ax,Ix,TP,etax,chix,thetax,pax)
c_____________________________________________________________________
COMMENT:  CONVERT ALL DEGREES TO RADIANS:
C
        theta = thetax*conv
          eta = etax*conv
          chi = chix*conv
           pa = pax*conv
C
COMMENT:  This 1st call to UPART calculates the C Matrix.
C         ****NO INCREMENTAL CHANGES ARE INTRODUCED HERE****
C
        part=1
c
c_____________________________________________________________________
        CALL UPART(a,i,theta,eta,chi,pa,tp,tpi,conv,c,
     $       delu,dthe,deta,dchi,dpa,du,pu,mu,det,sac,m,n,part)
c_____________________________________________________________________
                write(19,545) delu
                write(18,545) delu
                write(18,914) C(1),C(2),C(3),C(4)
                write(19,914) C(1),C(2),C(3),C(4)
```

```
COMMENT: The required matrix is [partial F/partial u].
C          This matrix is constructed a row at a time.
C          Any subsequent call to UPART is to build the Matrix PC
C          elements by rows. This is very IMPORTANT. UPART is
C          called twice more to fill this required PC matrix.
C
C        part = 2 denotes the 2nd partials are being computed.
c
comment:  Set up the initial PU matrix and populate it with zeros.
c         This matrix is used later but filled in by rows at a time.
c         This population of zeros is required to prevent
c         segmentation errors.     recall:  [PU] = [4x4].
c
        do 90 ii = 1,4
                do 92 jj = 1,4
                pu(ii,jj) = 0.d+00
 92             continue
 90     continue
c
comment:  This next section contains the necessary do loops to build
c          the PUA  abnd PUB matrices.  THe loop continues in the
c          next few pages.
c_____________________________________________________________________
```

```
C************************** DO LOOP **********************************
comment:  the k loop is responsible for computing PUA and PUB.
c
        part = 2
c
        do 1000 k = 1,2,1
           if(k.gt.2) go to 1000
           if(k.eq.1) du = dux
           if(k.eq.2) du = -dux
c
comment:  the n loop is responsible for generating the elements.
c
        do 2000 n = 1,4
c
        if(n.eq.1) then
           theta = thetax*conv + du
           eta   = etax*conv
           chi   = chix*conv
           pa    = pax*conv
        elseif(n.eq.2) then
           theta = thetax*conv
           eta   = etax*conv + du
           chi   = chix*conv
           pa    = pax*conv
        elseif(n.eq.3) then
           theta = thetax*conv
           eta   = etax*conv
           chi   = chix*conv + du
           pa    = pax*conv
        elseif(n.eq.4) then
           theta = thetax*conv
           eta   = etax*conv
           chi   = chix*conv
           pa    = pax*conv + du
        endif
```

```
C-------------------------------------------------------------------
COMMENT:  This next call to UPART in the LOOP generates the
elements.
C         if k = 1 then PUA is being filled.
C         if k = 2 then PUB is being filled.
c-------------------------------------------------------------------
        call UPART(a,i,theta,eta,chi,pa,tp,tpi,conv,c,
     $       delu,dthe,deta,dchi,dpa,du,pu,mu,det,sac,m,n_part)
c-------------------------------------------------------------------
c
 2000   continue
c
c
        if(k.eq.1) then
                write(18,562) k
                write(19,562) k
                do 40 ii = 1,4
                        do 42 jj = 1,4
                        pua(ii,jj) = pu(ii,jj)
 42                     continue
 40             continue
                        write(18,910)((pua(ii,jj),jj=1,4),ii=1,4)
                        write(19,910)((pua(ii,jj),jj=1,4),ii=1,4)
c
        elseif(k.eq.2) then
                write(18,564) k
                write(19,564) k
                do 44 ii = 1,4
                        do 46 jj = 1,4
                        pub(ii,jj) = pu(ii,jj)
 46                     continue
 44             continue
                        write(18,910)((pub(ii,jj),jj=1,4),ii=1,4)
                        write(19,910)((pub(ii,jj),jj=1,4),ii=1,4)
        endif
 1000   continue
C-------------------------------------------------------------------
```

```
COMMENT:  Now, get the difference [PUA - PUB]; divide by 2 x du.
C
        write(19,566) k
c
        do 21 ii = 1,4
          do 22 jj = 1,4
          diff(ii,jj) = pua(ii,jj) - pub(ii,jj)
 22       continue
 21     continue
c
          write(19,910)((diff(ii,jj),jj=1,4),ii=1,4)
c
c ____________________ Assemble the PC matrix._____________________
c
        write(19,568) k
c
        do 25 ii = 1,4
           do 26 jj = 1,4
           pc(ii,jj) = diff(ii,jj)/(2.d+00*du)
 26        continue
 25     continue
c
        write(19,910)((pc(ii,jj),jj=1,4),ii=1,4)
c
c               Calculate the INVERSE of matrix PC
c_____________________________________________________________________
        call INV4x4(pc,pci,det)
c_____________________________________________________________________
c
        write(19,912) det
c
comment: Terminate Job if matrix PC is singular.
c
        if(det.eq.0.0d+00) go to 999
c
        write(19,580)
        write(19,910)((pci(ii,jj),jj=1,4),ii=1,4)
```

```
        write(19,545) delu
        write(19,910) C(1),C(2),C(3),C(4)
C_____________________________________________________________________
COMMENT:  Determine the delta u requried and check for convergence.
C         If no convergence, repeat search for a zero by altering a
C         selected control parameter.
C
        write(*,590)
        write(19,590)
C
COMMENT: Intialize the DEL matrix to zero: [4x4] = [0]
C
        do 800 ii = 1,4
           del(ii) = 0.0d+00
 800    continue
C
COMMENT: Compute delta(u):  delta u = U(new) - U(old)
C                           delta u = del(i) = - C(1x4)*PCI[4x4]
C                           Unew    = Uold + delta u
C
COMMENT:  If deltau is not approx 0.0, then try it again with a
C         a slightly different choice of orientation angle.
C               Add the appropriate delta u to strat the iteration.
C
        do 801 ii = 1,4
           do 802 jj = 1,4
           del(ii) = -(C(jj)*pci(jj,ii) + del(ii))
           deldeg(ii) = del(ii)/conv
c
 802       continue
 801    continue
c



comment:  Print out the required change in u in degrees and radians.
```

```
c
        write(*,590)
        write(19,590)
        write(*,910)  del(1),del(2),del(3),del(4)
        write(19,910) del(1),del(2),del(3),del(4)
        write(*,592)
        write(19,592)
        write(*,910)  deldeg(1),deldeg(2),deldeg(3),deldeg(4)
        write(19,910) deldeg(1),deldeg(2),deldeg(3),deldeg(4)
c_____________________________________________________________________
comment: Determine if convergence is met.
c
c        if(del(1).lt.cv.and.del(2).lt.cv.and.del(3).lt.cv.
     $      and.del(4).lt.cv) then
c
          write(*,990)
          write(19,990)
        endif
C_____________________________________________________________________
COMMENT: Set up the iteration block:
C
            if(iter.eq.iterx) then
                theta = theta + del(1)
                eta   = eta   + del(2)
                chi   = chi   + del(3)
                pa    = pa    + del(4)
c
                thetax = thetax + deldeg(1)
                etax   = etax   + deldeg(2)
                chix   = chix   + deldeg(3)
                pax    = pax    + deldeg(4)
c
                call echo(ax,ix,tp,etax,chix,thetax,pax)
                call delta(a,i,mu,theta,eta,chi,pa,tpi,sac,tp,da)
c
                write(19,915) da
                write(*,915) da
```

```
c
                go to 999
              elseif(dec14.ne.0) then
                iter = iter + 1
c
comment:  echo the results of the run.
c
                call echo(ax,ix,tp,etax,chix,thetax,pax)
c
                 go to 209
              endif
 999     continue
c
        endfile(unit=19)
c
c_____________________ FORMATS _____________________
c
 100  FORMAT(1H1,/5X,'MOTION IN THE ECLIPTIC PLANE',5X,
     $'DATE:',i8,3x,'CASE:',i3,3x,'ITER:',i3,/)
 101  FORMAT(1H1,/5X,'MOTION IN THE NON-ECLIPTIC PLANE',5X,
     $'DATE:',i6,3x,'CASE:',i3,3x,'ITER:'i3,/)
 545     format(/39x,'Vector C [1x4]')
 562     format(/30x,'For k = ',i1,',  Matrix PUA  [4x4] ',/)
 564     format(/30x,'For k = ',i1,',  Matrix PUB  [4x4] ',/)
 566     format(/39x,'Matrix DIFF [4x4] ',20x,'k = ',i4,/)
 568     format(/39x,'Matrix PC  [4x4] ',20x,'k = ',i4,/)
 580     format(/39x,'Matrix PCI [4x4]',20x,'k = ',i4,/)
 590     format(/,'Vector DEL (rads) [1x4] ',/)
 592     format(/,'Vector DEL (degs) [1x4] ',/)
 910     format(4(d20.10))
 912     format(/5x,'Determinant = ',f50.40)
 914     format(/5x,f20.10,f20.10,f20.10,f20.10)
 915     format(/5x,'Da = ',f20.10)
 990     format(//5x,'By jove, I think you got it, Solar Sailor')
C
        STOP
        END
```

```
      SUBROUTINE UPART(a,i,theta,eta,chi,pa,tp,tpi,conv,c,
     $  delu,dthe,deta,dchi,dpa,du,pu,su,det,sac,s,n,part)
c
c       This subroutine calculates Matrix PU. It uses the DELTA
c       routine to calculate the changes in the orbital
c       parameters.
c
        integer s,n,part
        double precision a,i,sac,tp,tpi,su,conv,dif
        double precision theta,eta,chi,pa,dthe,deta,dchi,dpa
        double precision thetap,etap,chip,pap,delu,du
        double precision thetas,etas,chis,pas
        double precision det,dap,das
        double precision PU(4,4),C(4)
c_______________________________________________________________
comment: Initialize the da value to zero.
c
        dap = 0.d+00
        das = 0.d+00
c
comment: Get the orbital parameter perturbations for the given set
c       of orbital parameters and orientation angles. Use these for
c       the basis.
c
comment: This first call to DELTA computes the F(old) values.
c          We want to determine the perturbations without small
c          changes dthe,deta,etc. i.e., dthe=deta=dchi=dpa=0.0
c
comment:  The next page contains the DO LOOP for the construction of
c          components of the PUA and PUB matrices.  This constitutes
c          the n loop of the (s,n) LOOP started in the MAIN program.
c
```

```
c_____ DO LOOP _____
comment:  Determine the partials.  Start a do loop for the mth and
c         nth components of the matrix. This is done by rows.
c
        if(part.eq.1) m = 1
        do 400 n = 1,4
c
          if(n.eq.1) then
            dthe  = delu
            deta  = 0.d+00
            dchi  = 0.d+00
            dpa   = 0.d+00
          elseif(n.eq.2) then
            dthe  = 0.d+00
            deta  = delu
            dchi  = 0.d+00
            dpa   = 0.d+00
          elseif(n.eq.3) then
            dthe  = 0.d+00
            deta  = 0.d+00
            dchi  = delu
            dpa   = 0.d+00
          elseif(n.eq.4) then
            dthe  = 0.d+00
            deta  = 0.d+00
            dchi  = 0.d+00
            dpa   = delu
          endif
C
COMMENT: The following updates the angles in DELTA for PU matrix
c          determination. The suffix "p" (plus) means that the delu
c          is added when applicable.
c
            thetap = theta + dthe
            etap   = eta   + deta
            chip   = chi   + dchi
            pap    = pa    + dpa
```

```
comment:  This 1st to DELTA calculates the F(plus) values.
c-------------------------------------------------------------------
      call DELTA(a,i,mu,thetap,etap,chip,pap,tpi,sac,tp,dap)
c-------------------------------------------------------------------
c
comment:  This following section sets up the minus delu's.
c         It updates the angles in DELTA for PU determination.
c        The suffix m means that the delu is subtracted when
c        applicable. NOTE:  The dthe,deta,... are negative now.
c
      thetam = theta - dthe
      etam   = eta   - deta
      chim   = chi   - dchi
      pam    = pa    - dpa
c-------------------------------------------------------------------
comment:  This 2nd call to DELTA calculates the F(minus) values.
c-------------------------------------------------------------------
      call DELTA(a,i,mu,thetam,etam,chim,pam,tpi,sac,tp,dam)
c-------------------------------------------------------------------
c
comment:  The following sets up the approximations of each partial.
c          Print out these (dap - dam) values:
c          The following sets up Matrix PU [4x4].  Note the
c          denominator.This is 2 x delu because of the method of
c          approximation.
c
         dif = dap - dam
         if(part.eq.1) C(n) = dif/(2.d+00*delu)
         if(part.eq.2) pu(m,n) = dif/(2.d+00*delu)
c
comment:  start forming the elements of the PU matrices.
c               if k = 1, then PUA is being formed.
c               if k = 2, then PUB is being formed.
c
 400   continue
       return
       end
```

```
c-------------------------------------------------------------------------
      SUBROUTINE DELTA(a,i,mu,theta,eta,chi,pa,tpi,sac,tp,da)
c
comment:  This subroutine calculates the orbital parameter
c           perturbation due to changes in orientation angles
c           (theta,eta,chi,pa) and changes in orbital parameters.
c
        DOUBLE PRECISION D1,D2,D3,B1,B2,B3,B4,B5,B6,B7,B8
        DOUBLE PRECISION theta,eta,chi,pa,tp,tpi,a,i,mu,sac
        DOUBLE PRECISION da,da1,da2,da3,da4,da5,da6
C-------------------------------------------------------------------------
        B1 = dsin(theta)*dcos(chi)
        B2 = dsin(theta)*dsin(chi)*dcos(eta)
        B3 = dcos(theta)*dsin(chi)*dsin(eta)
        B4 = dsin(theta)*dsin(chi)
        B5 = dsin(theta)*dcos(eta)
        B6 = dcos(theta)*dcos(chi)*dsin(eta)
        B7 = dsin(theta)*dsin(eta)
        B8 = dcos(theta)*dcos(eta)
C-------------------------------------------------------------------------
        D1 = dsin(theta)*dcos(chi)*dcos(i)
        D2 = dsin(theta)*dsin(chi)*dcos(eta)*dcos(i)
     $     + dsin(theta)*dsin(eta)*dsin(i)
        D3 = dcos(theta)*dsin(chi)*dsin(eta)*dcos(i)
     $     - dcos(theta)*dcos(eta)*dsin(i)
c-------------------------------------------------------------------------
comment:  Calculate the following factors that enter into the Da
c           equation.

        da1 = D1**2.d+00/4.d+00*((B2 + 3.d+00*B4)*dcos(pa)
     $     - (3.D+00*B1 + B5)*dsin(pa))
        da2 = -D1*D2/2.D+00*((B2-B4)*dsin(pa) + (B5 - B1)*dcos(pa))
        da3 = D1*D3*2.d+00*(B6*dcos(pa) - B3*dsin(pa))
        da4 = D2*D3*2.d+00*(B3*dcos(pa) + B6*dsin(pa))
        da5 = D2**2.d+00/4.d+00*((3.d+00*B2 + B4)*dcos(pa)
     $     - (B1+3.d+00*B5)*dsin(pa))
        da6 = D3**2.d+00*((B2+B4)*dcos(pa) - (B1 + B5)*dsin(pa))
```

```
c
comment: Calculte the change in Semi-major axis.
c
        da = a**1.5d+00*sac*tp/mu*(da1+da2+da3+da4+da5+da6)
c
        write(18,581) da
 581    format(/5x,"Da = ", f20.10)
c
       return
       end
C
       SUBROUTINE ECHO(ax,ix,tp,etax,chix,thetax,pax)
C
C      THIS SUBROUTINE ECHOES INPUT TO SCREEN FOR VERIFICATION.
C
       DOUBLE PRECISION ax,ix,tp,etax,chix,thetax,pax
C      write(*,500)
       write(19,500)
       write(*,501)ax,ix,tp
       write(19,501)ax,ix,tp
       write(*,502)etax,chix,thetax,pax
       write(19,502)etax,chix,thetax,pax
C
C                    FORMATS
C
 500   FORMAT(10X,'YOUR SYSTEM IS AS FOLLOWS:')
C
 501   FORMAT(20X,'ORBITAL PARAMETERS:  a (DU)   =',D17.10,
     $/20X,'                      i  (DEG) =',D17.10,
     $/20X,'                      TP (TU)  =',D17.10)
 502   FORMAT(20X,'H ORIENTATION:   eta   (DEG)  =',D17.10,
     $/20X,'                    chi   (DEG)  =',D17.10,
     $/20X,'  CONING ANGLE:     theta (DEG)  =',D17.10,
     $/20X,'  PHASE ANGLE       pa    (DEG)  =',D17.10)
C
       RETURN
       END
```

```
C_____________________________________________________________________
      SUBROUTINE INV4X4(PC,PCI,det)
C
COMMENT:  This subroutine calculates the inverse of the [4x4] PC
C          matrix using CRAMER'S RULE. Primitive, but it works!
C_____________________________________________________________________
      double precision pc(4,4),pci(4,4),det
      double precision dp11,dp12,dp13,dp14,dp21,dp22,dp23,dp24
      double precision dp31,dp32,dp33,dp34,dp41,dp42,dp43,dp44
      double precision d1122,d1123,d1124,d1221,d1223,d1224
      double precision d1321,d1322,d1324,d1421,d1422,d1423
      double precision d2112,d2113,d2114,d2211,d2213,d2214
      double precision d2311,d2312,d2314,d2411,d2412,d2413
      double precision d3112,d3113,d3114,d3211,d3213,d3214
      double precision d3311,d3312,d3314,d3411,d3412,d3413
      double precision d4112,d4113,d4114,d4211,d4213,d4214
      double precision d4311,d4312,d4314,d4411,d4412,d4413
C_____________________________________________________________________
COMMENT: Determine the elements of each row of the determinant
C         and cofactor matrix. Data entered by column.
C_____________________________________________________________________
COMMENT:  1st row elements:
C_____________________________________________________________________
      d1122 = pc(3,3)*pc(4,4) - pc(4,3)*pc(3,4)
      d1123 = pc(3,2)*pc(4,4) - pc(4,2)*pc(3,4)
      d1124 = pc(3,2)*pc(4,3) - pc(4,2)*pc(3,3)
      dp11 = pc(2,2)*d1122 - pc(2,3)*d1123 + pc(2,4)*d1124
C_____________________________________________________________________
      d1221 = d1122
      d1223 = pc(3,1)*pc(4,4) - pc(4,1)*pc(3,4)
      d1224 = pc(3,1)*pc(4,3) - pc(4,1)*pc(3,3)
      dp12 = pc(2,1)*d1221 - pc(2,3)*d1223 + pc(2,4)*d1224
C_____________________________________________________________________
      d1321 = d1123
      d1322 = d1223
      d1324 = pc(3,1)*pc(4,2) - pc(4,1)*pc(3,2)
      dp13 = pc(2,1)*d1321 - pc(2,2)*d1322 + pc(2,4)*d1324
```

```
C-------------------------------------------------------------------
      d1421 = d1124
      d1422 = d1224
      d1423 = d1324
      dp14 = pc(2,1)*d1421 - pc(2,2)*d1422 + pc(2,3)*d1423
C-------------------------------------------------------------------
COMMENT:  2nd row elements:
C-------------------------------------------------------------------
      d2112 = pc(3,3)*pc(4,4) - pc(4,3)*pc(3,4)
      d2113 = pc(3,2)*pc(4,4) - pc(4,2)*pc(3,4)
      d2114 = pc(3,2)*pc(4,3) - pc(4,2)*pc(3,3)
      dp21 = pc(1,2)*d2112 - pc(1,3)*d2113 + pc(1,4)*d2114
C-------------------------------------------------------------------
      d2211 = d2112
      d2213 = pc(3,1)*pc(4,4) - pc(4,1)*pc(3,4)
      d2214 = pc(3,1)*pc(4,3) - pc(4,1)*pc(3,3)
      dp22 = pc(1,1)*d2211 - pc(1,3)*d2213 + pc(1,4)*d2214
C-------------------------------------------------------------------
      d2311 = d2113
      d2312 = d2213
      d2314 = pc(3,1)*pc(4,2) - pc(4,1)*pc(3,2)
      dp23= pc(1,1)*d2311 - pc(1,2)*d2312 + pc(1,4)*d2314
C-------------------------------------------------------------------
      d2411 = d2114
      d2412 = d2214
      d2413 = d2314
      dp24 = pc(1,1)*d2411 - pc(1,2)*d2412 + pc(1,3)*d2413
C-------------------------------------------------------------------
COMMENT:  3rd row elements:
C-------------------------------------------------------------------
      d3112 = pc(2,3)*pc(4,4) - pc(4,3)*pc(2,4)
      d3113 = pc(2,2)*pc(4,4) - pc(4,2)*pc(2,4)
      d3114 = pc(2,2)*pc(4,3) - pc(4,2)*pc(2,3)
      dp31 = pc(1,2)*d3112 - pc(1,3)*d3113 + pc(1,4)*d3114
C-------------------------------------------------------------------
```

```
C_____________________________________________________________
      d3211 = d3112
      d3213 = pc(2,1)*pc(4,4) - pc(4,1)*pc(2,4)
      d3214 = pc(2,1)*pc(4,3) - pc(4,1)*pc(2,3)
      dp32 = pc(1,1)*d3211 - pc(1,3)*d3213 + pc(1,4)*d3214
C_____________________________________________________________
      d3311 = d3113
      d3312 = d3213
      d3314 = pc(2,1)*pc(4,2) - pc(4,1)*pc(2,2)
      dp33 = pc(1,1)*d3311 - pc(1,2)*d3312 + pc(1,4)*d3314
C_____________________________________________________________
      d3411 = d3114
      d3412 = d3214
      d3413 = d3314
      dp34 = pc(1,1)*d3411 - pc(1,2)*d3412+  pc(1,3)*d3413
C_____________________________________________________________
COMMENTS:  4th row elements:
C_____________________________________________________________
      d4112 = pc(2,3)*pc(3,4) - pc(3,3)*pc(2,4)
      d4113 = pc(2,2)*pc(3,4) - pc(3,2)*pc(2,4)
      d4114 = pc(2,2)*pc(3,3) - pc(3,2)*pc(2,3)
      dp41 = pc(1,2)*d4112 - pc(1,3)*d4113 + pc(1,4)*d4114
C_____________________________________________________________
      d4211 = d4112
      d4213 = pc(2,1)*pc(3,4) - pc(3,1)*pc(2,4)
      d4214 = pc(2,1)*pc(3,3) - pc(3,1)*pc(2,3)
      dp42 = pc(1,1)*d4211 - pc(1,3)*d4213 + pc(1,4)*d4214
C_____________________________________________________________
      d4311 = d4113
      d4312 = d4213
      d4314 = pc(2,1)*pc(3,2) - pc(3,1)*pc(2,2)
      dp43 = pc(1,1)*d4311 - pc(1,2)*d4312 + pc(1,4)*d4314
C_____________________________________________________________
      d4411 = d4114
      d4412 = d4214
      d4413 = d4314
      dp44 = pc(1,1)*d4411 - pc(1,2)*d4412 + pc(1,3)*d4413
```

```
C_____________________________________________________________
COMMENT:  Calc the determinant of [PC].
C_____________________________________________________________
        det = pc(1,1)*dp11 - pc(1,2)*dp12 + pc(1,3)*dp13 -
       $        PC(1,4)*dp14
C
COMMENT: Flag the case when a singular matrix exist.,i.e., det= 0.0
C
        if(det.eq.0.d+00) then
          print*,'DANGER! DANGER! Det [PC] = 0.0'
          print*,'Program is stopped in Subroutine INV4X4:no output'
          go to 208
        endif
C_____________________________________________________________
COMMENT:  Calculate the Inverse [Pci] of [Pc].
C_____________________________________________________________
        pci(1,1) =  dp11/det
        pci(1,2) = -dp21/det
        pci(1,3) =  dp31/det
        pci(1,4) = -dp41/det
        pci(2,1) = -dp12/det
        pci(2,2) =  dp22/det
        pci(2,3) = -dp32/det
        pci(2,4) =  dp42/det
        pci(3,1) =  dp13/det
        pci(3,2) = -dp23/det
        pci(3,3) =  dp33/det
        pci(3,4) = -dp43/det
        pci(4,1) = -dp14/det
        pci(4,2) =  dp24/det
        pci(4,3) = -dp34/det
        pci(4,4) =  dp44/det
C
 208    continue
        return
        end
```

Appendix E

Optimization Test Case Output.

A test case was run to determine the required changes in an initial set of control parameter necessary to reach an stationary value for the Performance Index J. The specific input data with the resulting output are given below. For purpose of demonstration, two iterations are shown here. An option does exist for n iterations or until a certain predetermined convergence value is reached. Convergence is set at .001 radians or .057 degrees and is reached in two iterations.

Case Tested. The case (depicted by Figure 2.4D) is chosen as the test for the search scheme performance demostration. The following are the input data:

a - 2.0 du
i - 45.0°
θ - 50°
η - 0.0°
ζ - 0.0°
PA - -90°

From the Figure 2.4C, the local maximum appears to be about 55°. This is what is expected from the search scheme. As is shown later in the next few pages, the perspective provided a very good estimate of where the actual maximum is located with the initial conditions given to within .3 degrees. This is not always the case since pinpointing the maximum from the figures is limited to the amount of divisions in the two variables θ and PA. However, increments of 5 degrees is sufficient to make a good initial guess with. This is strongly exemplified by the test case which follows:

SYSTEM IS AS FOLLOWS:

ORBITAL PARAMETERS:	a (DU)	=	.2000000000e+01
	i (DEG)	=	.4500000000e+02
	tp (TU)	=	.1777153175e+02
ORIENTATION:	eta (DEG)	=	.0000000000e+00
	chi (DEG)	=	.0000000000e+00
CONING ANGLE:	theta (DEG)	=	.5000000000e+02
PHASE ANGLE:	pa (DEG)	=	-.9000000000e+02

Vector C [1x4]

.0000180033	.0000000000	.0000000000	.0000000000

For k = 1, Matrix PUA [4x4]

.6597125567e-05	.0000000000e+00	.0000000000e+00	.0000000000e+00
.1836358803e-04	.2719543322e-05	-.2156448749e-05	-.3691114090e-05
.1795098150e-04	-.2140999434e-05	-.3811105779e-05	.6321384922e-05
.1798084043e-04	-.3697125027e-05	.6322957922e-05	-.6323023559e-05

For k = 2, Matrix PUB [4x4]

.3087931771e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.1836358803e-04	-.2719543322e-05	.2156448749e-05	.3691114090e-05
.1795098150e-04	.2140999434e-05	.3811105779e-05	-.6321384922e-05
.1798084043e-04	.3697125027e-05	-.6322957922e-05	.6323023559e-05

Matrix DIFF [4x4]

-.2428219214e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	.5439086644e-05	-.4312897499e-05	-.7382228181e-05
.0000000000e+00	-.4281998867e-05	-.7622211558e-05	.1264276984e-04
.0000000000e+00	-.7394250055e-05	.1264591584e-04	-.1264604712e-04

Matrix PC [4x4]

.2428219214e-03	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	-.5439086644e-04	.4312897499e-04	.7382228181e-04
.0000000000e+00	.4281998867e-04	.7622211558e-04	-.1264276984e-03
.0000000000e+00	.7394250055e-04	-.1264591584e-03	.1264604712e-03

Determinant = -.000000000000000268858199652470203355747 5

Matrix PCI [4x4]

.4118244325e+04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	.5734031325e+04	.1335738514e+05	.1000663821e+05
.0000000000e+00	.1333372413e+05	.1114219114e+05	.3355638616e+04
.0000000000e+00	.9980849403e+04	.3331900100e+04	.5412247616e+04

Vector C [1x4]

.1800333992e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (rads) [1x4]

.7414215246e-01	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (degs) [1x4]

.4248032420e+01	.0000000000e+00	.0000000000e+00	.0000000000e+00

This is the end of the **INITIAL LOOP**.

ITERATION = 1

SYSTEM IS AS FOLLOWS:

ORBITAL PARAMETERS:	a (DU)	=	.2000000000e+01
	i (DEG)	=	.4500000000e+02
	TP (TU)	=	.1777153175e+02
ORIENTATION:	eta (DEG)	=	.0000000000e+00
	chi (DEG)	=	.0000000000e+00
CONING ANGLE:	theta (DEG)	=	.5424803242e+02
PHASE ANGLE	PA (DEG)	=	-.9000000000e+02

Vector C [1x4]

.0000016595	.0000000000	.0000000000	.0000000000

For k = 1, Matrix PUA [4x4]

-.7317446303e-05	.0000000000e+00	.0000000000e+00	.0000000000e+00
.2037513774e-05	.3816550277e-05	-.2443709575e-06	-.3230390882e-05
.1616555716e-05	-.2273911104e-06	-.3951553218e-05	.6356393714e-05
.1657456877e-05	-.3236146472e-05	.6358264047e-05	-.6358342092e-05

For k = 2, Matrix PUB [4x4]

.1230272911e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.2037513774e-05	-.3816550277e-05	.2443709575e-06	.3230390882e-05
.1616555716e-05	.2273911104e-06	.3951553218e-05	-.6356393714e-05
.1657456877e-05	.3236146472e-05	-.6358264047e-05	.6358342092e-05

Matrix DIFF [4x4]

-.1962017541e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	.7633100554e-05	-.4887419150e-06	-.6460781764e-05
.0000000000e+00	-.4547822207e-06	-.7903106436e-05	.1271278743e-04
.0000000000e+00	-.6472292943e-05	.1271652809e-04	-.1271668418e-04

Matrix PC [4x4]

.1962017541e-03	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	-.7633100554e-04	.4887419150e-05	.6460781764e-04
.0000000000e+00	.4547822207e-05	.7903106436e-04	-.1271278743e-03
.0000000000e+00	.6472292943e-04	-.1271652809e-03	.1271668418e-03

Determinant = .000000000000000010980981480027645913466 5

Matrix PCI [4x4]

.5096794392e+04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	-.1092792724e+06	-.1579012964e+06	-.1023329720e+06
.0000000000e+00	-.1573479727e+06	-.2481494526e+06	-.1681319044e+06
.0000000000e+00	-.1017271824e+06	-.1677808467e+06	-.1081826908e+06

Vector C [1x4]

.1659530858e-05	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (rads) [1x4]

.8458287572e-02	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (degs) [1x4]

.4846241798e+00	.0000000000e+00	.0000000000e+00	.0000000000e+00

This is the end of the **FIRST ITERATION.**

ITERATION = 2

SYSTEM IS AS FOLLOWS:

ORBITAL PARAMETERS:	a (DU)	=	.2000000000e+01
	i (DEG)	=	.4500000000e+02
	TP (TU)	=	.1777153175e+02
ORIENTATION:	eta (DEG)	=	.0000000000e+00
	chi (DEG)	=	.0000000000e+00
CONING ANGLE:	theta (DEG)	=	.5473265660e+02
PHASE ANGLE	PA (DEG)	=	-.9000000000e+02

Vector C [1x4]

.0000000211	.0000000000	.0000000000	.0000000000

For k = 1, Matrix PUA [4x4]

-.8664824950e-05	.0000000000e+00	.0000000000e+00	.0000000000e+00
.4001748609e-06	.3944497162e-05	-.1718988786e-07	-.3173732881e-05
-.2087678209e-07	-.3857831379e-10	-.3965840001e-05	.6356706917e-05
.2105726398e-07	-.3179454671e-05	.6358611424e-05	-.6358690894e-05

For k = 2, Matrix PUB [4x4]

.1038993518e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.4001748609e-06	-.3944497162e-05	.1718988786e-07	.3173732881e-05
-.2087678209e-07	.3857831379e-10	.3965840001e-05	-.6356706917e-05
.2105726398e-07	.3179454671e-05	-.6358611424e-05	.6358690894e-05

Matrix DIFF [4x4]

-.1905476013e-04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	.7888994325e-05	-.3437977571e-07	-.6347465762e-05
.0000000000e+00	-.7715662758e-10	-.7931680002e-05	.1271341383e-04
.0000000000e+00	-.6358909343e-05	.1271722285e-04	-.1271738179e-04

Matrix PC [4x4]

.1905476013e-03	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	-.7888994325e-04	.3437977571e-06	.6347465762e-04
.0000000000e+00	.7715662758e-09	.7931680002e-04	-.1271341383e-03
.0000000000e+00	.6358909343e-04	-.1271722285e-03	.1271738179e-03

Determinant = .000000000000000029876155962369095986375l

Matrix PCI [4x4]

.5248032476e+04	.0000000000e+00	.0000000000e+00	.0000000000e+00
.0000000000e+00	-.3878353975e+05	-.5176275317e+05	-.3238906549e+05
.0000000000e+00	-.5156189341e+05	-.8973109370e+05	-.6396766162e+05
.0000000000e+00	-.3216881292e+05	-.6384770652e+05	-.3990851931e+05

Vector C [1x4]

.2108361300e-07	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (rads) [1x4]

.1106474857e-03	.0000000000e+00	.0000000000e+00	.0000000000e+00

Vector DEL (degs) [1x4]

.6339633946e-02	.0000000000e+00	.0000000000e+00	.0000000000e+00

****

By jove, I think you got it, Solar Sailor!!!!!

****

FINAL SYSTEM IS AS FOLLOWS:

ORBITAL PARAMETERS:	a (DU)	=	.2000000000e+01
	i (DEG)	=	.4500000000e+02
	tp (TU)	=	.1777153175e+02
ORIENTATION:	eta (DEG)	=	.0000000000e+00
	chi (DEG)	=	.0000000000e+00
CONING ANGLE:	theta (DEG)	=	.5473899623e+02
PHASE ANGLE	pa (DEG)	=	-.9000000000e+02

This is the end of the **SECOND ITERATION.**

REMARKS: This test case showed that when the search point was initiated at theta = 50°, the search algorithm converged at a local maximum at theta = 54.73°. From Figure 4.2C, this can be visually verified. The close agreement between the perspectives and the computed maximum gives strong credence in the method of search and the utility of the perspectives in providing a good initial guess. This last control vector set was inputted into the DELTA A function to verify the finding within the theta equal 54.7°. The results are tabulated below for a fix inclination at 45°:

	THETA	DELTA A ($\times 10^{-4}$ DU)	
	54.0	1.272 129	
	54.1	1.272 170	
	54.2	1.272 205	
	54.3	1.272 233	
	54.4	1.272 256	
	54.5	1.272 273	
	54.6	1.272 284	
--→	54.7	1.272 289	←--
	54.8	1.272 288	
	54.9	1.272 281	
	55.0	1.272 269	

The search scheme is more accurate in pin-pointing the exact maximum coordinate. The coordinate is essentially the control vector set that would optimize the change in semi-major axis. Note, however, that other maxima do exist and can easily be located with the aid of the corresponding perspective and /or with the search algorithm. Care must be excercised when doing so; the perspectives give the angular momentum orientation angles as zero degrees. The search algorithm will find maxima without holding these angles fixed. All variables are incremented. Plotting the perspective for a particular output will provide an extra dimension in locateing and verifying the computed maxima.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

AD-A153175

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GA/AA/84D-2	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Air Force Institute of Technology; 6b. OFFICE SYMBOL (If applicable): AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.

11. TITLE (Include Security Classification): Maximization of the Semi-major Axis of a Freely Coning Solar Sail

12. PERSONAL AUTHOR(S): Mario R. Borja, Capt USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	December, 1984	137

16. SUPPLEMENTARY NOTATION

Approved for public release: IAW AFR 190-17
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB, OH 45433

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
03	03	
22	03	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Solar Sails, Orbital Perturbations, Coning Motion Optimization

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This study addresses the maximization of the circular orbit of an earth-orbiting solar sail under free coning motion. The objective is to find the optimal sail settings which produce the most change in the semi-major axis per orbit. Angular orientations ~~wrt~~ sail nutation, precession, mean motion, and its angular momentum control the magnitude of the solar thrust along the Sailcraft's velocity direction. A numerical search scheme uses the central differencing iteration method to identify sets of control parameters meeting certain optimality conditions that produce a stationary value in a selected performance index. Such scheme displays it vulnerability to a lack of a good initial guess. Three-dimensional perspectives of the small perturbation equation describing the behavior of the change in semi-major axis facilitates the understanding of it cyclic nature and provides an excellent tool for identifying the various areas/locations of possible maxima and...

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. William Wiesel, Jr.		AFIT/ENY

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

minima as well as slope-critical areas. A test case demonstrates the location of particular points of interest with few search iterations. The study shows the need to cross reference the search output to the respective perspective.

END

FILMED

5-85

DTIC